Inverse Modelling and Inverse Simulation for System Engineering and Control Applications

Linghai Lu

Department of Electronics and Electrical Engineering

University of Glasgow

A thesis submitted for the degree of

Doctor of Philosophy

to the University of Glasgow

 October 2007

Abstract

Following extensive development over the past two decades, techniques of inverse simulation have led to a range of successful applications, mainly in the fields of helicopter flight mechanics, aircraft handling qualities and associated issues in terms of model validation. However, the available methods still have some well-known limitations. The traditional methods based on the Newton-Raphson algorithm suffer from numerical problems such as high-frequency oscillations and can have limitations in their applicability due to problems of input-output redundancy. The existing approaches may also show a phenomenon which has been termed “constraint oscillations” which leads to low-frequency oscillatory behaviour in the inverse solutions. Moreover, the need for derivative information may limit their applicability for situations involving manoeuvre discontinuities, model discontinuities or input constraints.

Two new methods are developed to overcome these issues. The first one, based on sensitivity-analysis theory, allows the Jacobian matrix to be calculated by solving a sensitivity equation and also overcomes problems of input-output redundancy. In addition, it can improve the accuracy of results compared with conventional methods and can deal with the problem of high-frequency oscillations to some extent. The second one, based on a constrained Nelder-Mead search-based optimisation algorithm, is completely derivative-free algorithm for inverse simulation. This approach eliminates problems which make traditional inverse simulation techniques difficult to apply in control applications involving discontinuous issues such as actuator amplitude or rate limits.

This thesis also offers new insight into the relationship between mathematically based techniques of model inversion and the inverse simulation approach. The similarities and shortcomings of both these methodologies are explored. The findings point to the possibility that inverse simulation can be used successfully within the control system

design process for feedforward controllers for model-based output-tracking control system structures. This avoids the more complicated and relatively tedious techniques of model inversion which have been used in the past for feedforward controller design.

The methods of inverse simulation presented in this thesis have been applied to a number of problems which are concerned mainly with helicopter and ship control problems and include cases involving systems having nonminimum-phase characteristics. The analysis of results for these practical applications shows that the approaches developed and presented in this thesis are of practical importance. It is believed that these developments form a useful step in moving inverse simulation methods from the status of an academic research topic to a practical and robust set of tools for engineering system design.

Acknowledgements

First and foremost, I am extremely grateful to my direct supervisor Professor D.J. Murray-Smith for his generous and wise guidance, penetrating criticism, inspiration, encouragement, and good company during the three-year development of my PhD research. Without many insightful conversations with him and the consequent instructions from him during the development of the ideas in this thesis, I would have been lost. I am also grateful to Dr E.W. McGookin for his supervision at each stage of the work with helpful comments and suggestions. He always was available when I needed his advice

I also thank Dr D.G. Thomson in the Department of Aerospace Engineering for the many useful discussions that I had with him as well as for providing resources and constructive ideas during the development of the sensitivity analysis method for inverse simulation and its final successful publication. In addition, I wish to thank Dr Marat Bagiev, of the Department of Aerospace Engineering of the University of Glasgow, for supplying the Lynx helicopter model used in this thesis.

Next thanks must be given to all the friends that I have made during my period in Glasgow, in particular Mr Tom O'hara, Kevin Worrall, Rosdiazli Ibrahim, David McGeoch, Yang Li, and Meghan Loo, for their friendship, sincere help, and moral support. I also thank for Yawei Wu for her patience and encouragement. In addition, I gratefully acknowledge the award of a University of Glasgow Scholarship and an Overseas Research Studentship from the British Government.

Finally, I am forever indebted to my parents Mei Lu and Liming Ye for their understanding, their love, and encouragement when it was most required. I cannot end without thanking my family,

Contents

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

List of Tables

Abbreviation

2DOF	Two Degrees-of-Freedom
AUV	Autonomous Underwater Vehicle
BFGS	Broyden, Fletcher, Goldfar and Shanno
FBC	Feedback Controller
FFC	Feedforward Controller
FFS	Feedforward Simulation
LFT	Linear Fractional Transformation
LHP	Left-Half Plane
LM	Levenberg-Marquardt
LQ	Linear Quadratic
MECP	Energy Control Problem
MI	Model Inversion
MIMO	Multiple-Input Multiple-Output
MP	Minimum-Phase
MTE	Mission Task Element
NR	Newton-Raphson
NM	Nelder-Mead
NMP	Nonminimum-Phase
PDE	Partial Differential Equation
PID	Proportional-Integral-Derivative
RHP	Right-Half Plane
RK	Runge-Kutta
RMS	Root-Mean-Square
RRS	Rudder-Roll Stabilisation System
RZ	Rov Zeefakkel

SA	Sensitivity Analysis
SISO	Single-Input-Single-Output
SNR	Signal-to-Noise Ratio
UAV	Unmanned Aerial Vehicle

Chapter 1

Introduction

Contents

This chapter provides general background information about the research carried out, the context in which the project was defined, the contributions made during the research period and an outline of the thesis.

1.1 Background

Conventional forward simulation using computers is an established part of the process of modelling of systems through mathematical models. The process is used to attempt to find solutions that predict the behaviour of the system (usually as a function of time) from a given model structure, a set of parameters and a set of initial conditions. In the aerospace and marine fields, this simulation process usually involves the determination of the motions of vehicles in response to applied inputs (Hess, Gao, & Wang, 1991).

This process of forward modelling and simulation is not, however, the only approach possible for gaining an understanding of the behaviour of complex engineering systems. The idea inverting a system can be traced back to the 1930s when Jones (1936) investigated the effects of gusts on aircraft by inverting a linearised aircraft model. In the 1960s and

1970s of the last century, a class of methods termed dynamic inversion or feedback linearization were developed for multivariable nonlinear minimum-phase (MP) systems by some pioneers (Brockett, 1965; Dorato, 1969; Hirschorn, 1979; Sain & Massey, 1969; Silverman, 1969). The procedure of dynamic inversion involves the transformation of the original nonlinear system into a linear and controllable system via a nonlinear state space change of coordinates and a nonlinear state feedback control law with the application of differential geometry concepts. Subsequently, further efforts were made to develop approaches for nonminimum-phase (NMP) systems (Isidori, 1989; Isidori & Byrnes, 1990) and in the mid 1990s, Devasia, Chen and Paden (1996) achieved further significant progress by developing a method based on noncausal *Picard-like* iteration to obtain bounded inversions. In recent years, based on this work, several approaches have been developed to achieve the causal inversion of NMP systems (Wang & Chen, 2001; 2002b; Zou & Devasia, 1999; 2007) as well as methodology for the nonhyperbolic problem (Devasia, 1997; 1999).

This group of approaches has attracted such considerable attention due to the fact that these techniques offer a potentially powerful methodology for control system design. Reiner, Balas and Garrard (1995) designed an outer-loop flight controller with structured singular value (μ) synthesis, based on the linearised model obtained by an inner-loop controller designed by dynamic inversion. Hu *et al* (2003) followed the similar approach to design a controller for ship course keeping.

In particular, one of the main stimuli to the development of these methods is that dynamic inversion can be used to design an inversion-based feedforward controller (FFC) to obtain precision tracking of a particular output trajectory (Devasia, 1997). One of the reasons for this development relates to the possibilities that it offers for solving some problems encountered by the traditional regulator approach, such as the transient tracking error after the switching instants (Qui & Davison, 1993). However, it has been pointed out that the transient tracking error problem can be eliminated by the stable-inversion based approach (Devasia & Paden, 1998; Hunt & Meyer, 1997).

In addition, the specification of trajectories can be simplified and, furthermore, dynamic inversion avoids the need for implementation of gain scheduling, such as to ensure flight control system stability over the entire operational envelope of the aircraft or other controlled system (Devasia, 1997).

However, all of the above approaches have some shortcomings that restrict their application. Firstly, all the available mathematical approaches involving model inversion are quite tedious and difficult to implement in practice even for MP systems, especially in applications involving high-order models, such as those encountered in aircraft flight control or in the marine field where there the number of state variables can be large. Secondly, most available methods only achieve noncausal inversion and depend on the whole desired output trajectory. The causal approach (Wang & Chen, 2001; 2002b) eliminates the issue of noncausality but at the cost of a reduction of the tracking accuracy and an increase of complexity in implementation. Thirdly, the methods for model inversion usually require sufficient smoothness of the desired trajectories to provide the ideal input vector constructed by following the vector relative degree (Appendix-A). However, in practice, the assumption of the existence of the strictly relative degree can sometimes be violated (Sastry, 1999). Ramakrishna, Hunt and Meyer (2001) suggest a solution for some kinds of violations but this can be difficult to implement. Finally, the valid domain of these methods is usually restricted to tracking of trajectories involving small amplitudes due to the form of the algorithms (Sastry, 1999). Therefore, they may be not suitable for more demanding or severe types of manoeuvres.

During the period when major developments were taking place in terms of the techniques for model inversion, a numerical process termed inverse simulation was being developed to allow numerical inversion of linear and nonlinear dynamic models. This approach has attracted significant attention within the field of aerospace engineering and in some other application areas. It is an approach that generates the forward control inputs such that a mathematical model of a system can follow a prescribed trajectory in state space. Many contributions have been published with the aim of developing more numerically stable and robust approaches to inverse simulation (Avanzini, de Matteis & de Socio, 1999; Celi,

2000; de Matteis, de Socio & Leonessa, 1995; Hess *et al*, 1991; Kato & Saguira, 1986; Lee & Kim, 1997; Lu, Murray-Smith & Thomson, 2007c; Thomson, 1987).

The reasons for the popularity of inverse simulation methods arise from the practical usefulness of this approach in various fields. Firstly, Thomson and Bradley (1990b; 1994) highlighted the value of inverse simulation techniques in the investigation of the handling qualities, manoeuvrability, and agility of a hypothetical battlefield utility helicopter represented by a set of configurational data at the conceptual design stage. The quantitative assessment of helicopter handling qualities and validation of the model was approached by analysing attitude quickness criteria through simulated flight involving standard Mission Task Elements (MTEs) (Bradley & Thomson, 1993; Rutherford & Thomson, 1997; Thomson & Bradley, 1998). Secondly, inverse simulation has been shown to facilitate investigation of control actions required following engine failures during takeoff from offshore platforms (Thomson *et al*., 1995). In addition, inverse simulation has been investigated for output-tracking and inversion-based controllers (Avanzini *et al*., 1999; Boyle & Chamitoff, 1999; Gray & Grünhagen, 1998, Sentoh & Bryson, 1992).

As with the model inversion approaches, traditional inverse simulation techniques suffer from some problems. Numerical issues, such as non-convergence, rounding error, and phenomena involving sustained high-frequency oscillations have been found (Avanzini & de Matteis, 2001; Hess *et al*., 1991; Lin, 1993; Rutherford & Thomson; 1996; 1997). Secondly, redundancy issues associated with the number of inputs being greater than the number of outputs may also lead to non-convergence of the solutions (de Matteis *et al*., 1995; Lee & Kim, 1997; Yip & Leng, 1998). Thirdly, in some cases oscillations of much lower frequency also appear in the results. This phenomenon, which has been given the term “constraint oscillations”, often gives stable results but with gently damped oscillating components (Anderson, 2003; Thomson & Bradley, 1990). Fourthly, the numerical processes in traditional methods of inverse simulation involve use of derivative information such as in the Jacobian matrix or the Hessian matrix. This limits their application only to smooth trajectories and models which have no input constraints. Finally, the stability of inverse simulation methods for NMP systems has not been intensively investigated. In

addition, there has been no previous consideration of the relationship between model inversion and inverse simulation techniques.

The general aims and initial objectives of the current research were to explore and investigate the problems existing in available techniques for inverse simulation and then to develop new methodologies to overcome some of these existing problems. In addition, an aim of this research was to investigate the applicability of inverse simulation to the nonlinear MP and NMP systems and to develop robust tracking controllers based on the traditional output-tracking control system structure with inverse simulation, thus avoiding the more complicated techniques of model inversion. Helicopter and marine systems were to be considered for these control applications.

1.2 Original contribution of research

To best of the author's knowledge the novel contributions made in this thesis are as shown in the following list:

- Development of the SA based method for inverse simulation to overcome: a) problems of high-frequency oscillations in inverse numerical solutions; b) the approximations involved in traditional methods for calculation of the Jacobian matrix, and c) the redundancy problem.

- Investigation of the applicability of inverse simulation to MP and NMP systems. A methodology has been developed for application of inverse simulation to the NMP system by dividing the inverse simulation process into two distinct steps.

- Development of a constrained and completely derivative-information-free NM-based method for inverse simulation.

- Investigation for the first time of the relationship between inverse simulation and model inversion methods.

- Investigation of the constraint-oscillation phenomenon in inverse simulation in more detail than in previous investigations.

- Investigation, for the first time, of the use of the inverse simulation methods on models involving input constraints and a combination of input constraints and discontinuous manoeuvres.

- Applications of inverse simulation involving problems from the marine field for the first time.

- Developments of the FFC using inverse simulation techniques within the 2DOF control scheme for the helicopter and ship control problems.

- Development of a new approach to calculate the Jacobian matrix by calculating the relative degree vector. Through use of this approach the numerical convergence processes of inverse simulation can be improved and become more stable and the problem of high-frequency oscillations can be eliminated.

The following conference and journal publications have been written during the course of this research:

1. Lu L, Murray-Smith DJ, and McGookin EW (2006). Applications of inverse simulation within the model-following control structure. *Proc. of International Control Conference*; ICC2006, Aug. 30th – Sept 1st, Glasgow, UK.

2. Lu L, Murray-Smith DJ, and McGookin EW (2006). Relationships between model inversion and inverse simulation techniques. *Proc. of 5th MATHMOD*; Feb 8-10, Vienna, Austria.

3. Lu L, Murray-Smith DJ, and McGookin EW (2007). Feedforward controller design from a constrained derivative-free inverse simulation process. *Control Engineering Practice*. Submitted for Publication.

4. Lu L, Murray-Smith DJ, and McGookin EW (2007). Investigation of inverse simulation for design of feedforward controllers. *Journal of Mathematical and Computer Modelling of Dynamical Systems*. Accepted for Publication.

5. Lu L, Murray-Smith DJ, and Thomson DG (2007). A sensitivity-analysis method for inverse simulation. *Journal of Guidance, Control, and* Dynamics, **30**(1), 114-121.

6. Lu L, Murray-Smith DJ, and Thomson DG (2007d). Issues of numerical accuracy and stability in inverse simulation. *Simulation Modelling Practice and Theory*. Under Revision.

1.3 Outline of thesis

Chapter 2 outlines the general direction of the research described in this thesis. It provides a brief history of the development of model inversion techniques and inverse simulation approaches. The potential disadvantages and advantages of these approaches are illustrated in detail to help explain the motivation for the work. This chapter presents the main relevant references that are applicable to the research area of this thesis as well.

Chapter 3 includes an investigation of the close relationship between model inversion and inverse simulation techniques. This investigation provides new insight concerning some aspects of inverse simulation and shows that inverse simulation techniques can provide an alternative way to deal with the implementation problems encountered with model inversion techniques for some situations. Cases considered in this context include the NMP and nonhyperbolic cases, involving a division of the inverse simulation process into two stages. The results from the analysis in this chapter, together with the case studies that are

implemented within it, provide a comprehensive background to the investigation of the possible use of inverse simulation methods to replace model inversion techniques in the design of feedforward controllers (FFCs).

Chapter 4 is devoted to an explanation of problems associated with processes involved in the implementation of inverse simulation, such as high-frequency oscillations, redundancy problems and constraint oscillations. In this context the main efforts described here relate to constraint oscillations. Effects of sampling rate, the type of manoeuvres undertaken, and the internal dynamics of the model itself are all taken into consideration in investigation of this phenomenon. In addition, an alternative method for calculating the vector relative degree is proposed and this approach also has relevance for calculation of the Jacobian matrix.

Chapter 5 presents a new formulation of the inverse simulation problem based on the theory of sensitivity-analysis (SA). This development allows the Jacobian matrix to be calculated by solving a sensitivity equation which may offer advantages over conventional methods of calculation based on approximation methods. The SA approach also makes full use of information within the time interval over which key quantities are compared in the inverse simulation algorithm and therefore can provide more accurate results. This technique also accounts for the issues of input-output redundancy that arise in the traditional approaches to inverse simulation. The results of a case study involving a nonlinear HS125 aircraft model and a Lynx helicopter model show improved performance in terms of stability and accuracy in comparison with the traditional Newton-Raphson (NR) approach to inverse simulation.

Chapter 6 presents a new methodology in which the inverse simulation problem is formulated as an optimisation problem based on the constrained derivative-free Nelder-Mead (NM) algorithm. Because of the absence of derivative information in the inverse simulation process, unlike the traditional approaches, the new technique enhances the feasibility and flexibility of the inverse simulation process. It also extends the applicability to situations involving discontinuous manoeuvres or input constraints and discontinuities within the model. Case studies involving ship models with input saturation of rudder angle, rudder rate or propeller speed or involving a discontinuous manoeuvre show that the

convergence and numerical stability properties of the inverse simulation based on the NM approach are generally superior to traditional approaches.

Chapter 7 discuses some theoretical aspects and practical implementations of the mixed-sensitivity (K/KS) H_∞ control methodology to design the feedback controller (FBC) in the two degrees-of-freedom (2DOF) scheme. The stability and robustness of the controllers in the presence of measurement noise and disturbances are investigated through applications involving a Norrbin type of nonlinear ship model, a linear Lynx-like helicopter model, and a highly detailed nonlinear Container ship model.

Chapter 8 describes the developments that make it possible to use inverse simulation in place of model inversion techniques to develop robust feedforward tracking controllers for the traditional 2DOF output-tracking control system structure. This chapter is of particular importance in that it includes the most significant findings of the whole project. It includes discussion of the relationship between the FFC and uncertainty and incorporates two illustrative case studies involving applications.

Chapter 9 describes the use of inverse simulation techniques within a helicopter guidance and control system for shipboard landing. The investigations are carried out both in still air and in the presence of wind-disturbances. A simplified approach is adopted to make the inclusion of the wind disturbance more accessible and easily applicable within the helicopter model. Therefore only four categories of wind disturbance are considered in terms of the wind direction. They are: tailwind, headwind, port wind, and starboard wind. In addition, this chapter includes the design and analysis of a control system for a ship landing manoeuvre based on the 2DOF control scheme.

Chapter 10 presents the main conclusion of the thesis and highlights the original contributions. In addition, suggestions are made for future work to build on the foundation established here.

Chapter 2

Fundamentals of Model Inversion and Inverse Simulation

Contents

This chapter presents a study of the historic development of model inversion techniques and inverse simulation approaches. It is hoped that this overview can provide insight regarding the advantages and disadvantages of the various different methods of approach and that this, in turn, can lead to new developments in the most promising directions.

2.1 The two degrees-of-freedom control scheme

In control applications, tracking control and regulation are two major researching topics and have thus attracted considerable attention from control researchers. Traditionally, asymptotic tracking has been solved for a given reference trajectory in the context of linear quadratic optimal control. Since the introduction of regulator theory for linear systems by Francis and Wonham (1976) and then further application to the nonlinear regulator field by Isidori and Byrnes (1990), output tracking through the 2DOF approach has undergone

significant development. Typically this control scheme involves two controllers, as shown in Fig. 2.1:

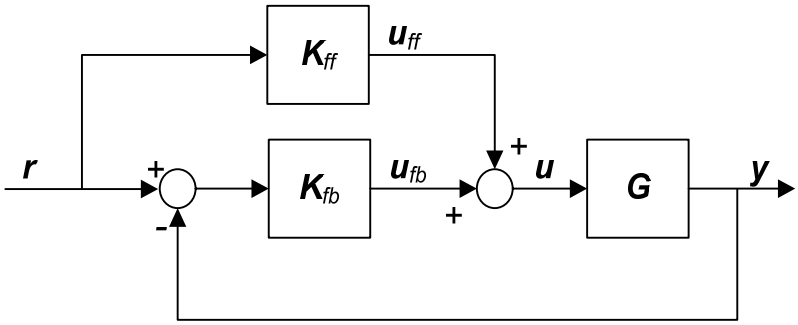

Fig. 2.1 Two degrees-of-freedom control design scheme (Graichen *et al.*, 2005)

where $\boldsymbol{K}_{ff}$ is the FFC and $\boldsymbol{K}_{fb}$ is the FBC. The FBC can be designed to guarantee robust stability and improves disturbance rejection, while the FFC improves tracking performance. A lot of techniques have published for the design of the FFC and FBC and satisfying tracking performance has been achieved in a number of applications, with reduced control energy requirements, and good robust stability (Braatz *et al*., 2006; Devasia *et al*., 1996; Lim & Chan, 2003; Muramatsu & Watanabe, 2005). Generally, the available approaches to design of this form of system can be split into two groups according to whether the two controllers are synthesized simultaneously or separately.

Limebeer, Kasenally and Perkins (1993) proposed an effective method, which is the extension of the H_∞ loop-shaping design procedure of McFarland and Glover, to design the FFC and FBC jointly. Although they are synthesized jointly as a controller, the roles of the FFC and FBC are quite distinct. The feedback part of the controller obtained in this way has the same roles as the traditional FBC, such as meeting given requirements in terms of robust stability and disturbance rejection. In addition, the feedforward part of the controller acts as the FFC to drive the closed-loop to follow a reference model. This development has been successfully illustrated in an example involving an application to an aero engine (Skogestad & Postlethwaite, 1996). In a recent development of this class of techniques

Wik, Fransson and Lennartsson (2003) present a performance optimization method for designing the FFC and FBC together for given control objectives with uncertainties.

The fact that the role that each controller has is distinct, as mentioned above, suggests the possibility of a two-stage design in which the FFC and FBC are designed separately (Giusto & Paganini, 1999). This approach is attractive because the procedure for the joint approach to the design of 2DOF systems is quite complex, whereas the separated process is relatively simple and straightforward. These benefits may compensate for disadvantages arising in the application of this decomposition procedure due to the fact that the FBC plays the principal role of solving the uncertainty and this may have an influence in terms of the achievable robust tracking performance for systems with uncertainties.

Due to the relatively straightforward synthesis scheme, a number of different approaches for the separated procedure have been developed. Burdisso and Fuller (1994) designed the FFC by eigenvalue assignment. Huang, Shah and Miller (2000) developed an approach to access the potential benefit of implementing FFC in an industrial process from routine operating data. A numerical example included in the published work on that approach shows the improved performance that can be achieved by adding the FFC. Mammar, Koenig and Nouveliere (2001) implemented the H_∞ algorithm to design both the FFC and FBC. In their approach the FBC is designed using robust interpolation of state feedback with loop-shaping controller scheduling. The robust FFC is designed by the Linear-Fractional-Transformation (LFT) method, similar to the H_∞ method, while the disturbance acts as the input and the input and output are directly considered. Moreover, the presented approach does not involve any restrictions such as requirements of the model being NMP.

Among the control system design methodologies relating to the separated procedure, inversion of system dynamics for FFC design has been the most widely investigated by control researchers in the last two decades, especially in the aircraft field (Mickle, Huang & Zhu, 2004; Moghaddam & Moosavi, 2005; Zou and Devasia, 2007), while the FBC is designed by a traditional control algorithm such as PID (Visioli, 2004; von Grunhagen *et al*., 1996), the H_∞ algorithm (Che & Chen, 2001; Takahashi, 1994), or the linear quadratic

(LQ) algorithm (Al-Hiddabi & McClamroch, 2002; Escande, 1997). Developments in this field have been stimulated by recognition of the defects existing in the regulator approach. Model-inversion based techniques have also been applied to other fields such as flexible structures and manipulators (Wang & Chen, 2002a) and in marine system control (Loo, McGookin & Murray-Smith, 2005). In addition, a new scheme, termed adaptive 2DOF control, is also applicable. This is based on feedback error learning where the FFC is tuned to be the inverse of the plant model (Muramatsu & Watanabe, 2004). The FFC can be described by an adaptive inverse model and it asymptotically tends to the inverse of the plant as time tends to infinity.

Therefore, the essential problem of the inversion-based 2DOF scheme is to find a suitable approach for inversion of system dynamics to design a FFC.

2.2 Review of methods for the inversion of nonlinear system dynamics

The details relating to techniques for dynamic inversion are reviewed in this section. The contents cover NMP systems and nonhyperbolic systems as well as other systems.

2.2.1 Noncausal inversion of nonlinear system dynamics

2.2.1.1 Inversion of nonminimum-phase systems

Some pioneers (e.g. Brockett, 1965; Dorato, 1969; Sain & Massey, 1969; Silverman, 1969) in the sixties of the last century showed that the dynamics of a linear system could be inverted to find inputs that exactly track a desired output trajectory. Hirschorn (1979) presented the set of sufficient conditions for the invertibility of multivariable nonlinear control systems. While the exact and asymptotic tracking results were obtained, all these previous approaches were based on MP control systems only. They led to unbounded inputs resulting from the unstable internal systems existing in NMP systems. Therefore, they could not be used for practical output tracking under this kind of situation. The same

challenging problem can be met in the regulation and path-following fields (Aguiar, Hespanha & Kokotovic, 2005; Al-Hiddabi & McClamroch, 2002).

Compared with MP systems, tracking for NMP systems is much more difficult. The limitations introduced by unstable zero dynamics are structural. They cannot be avoided without changing the system structure or reformulating the tracking problem. A major breakthrough in defining a general framework for tracking for nonlinear NMP systems was made by Devasia *et al* (1996) who developed and proved the convergence of a noncausal *Picard-like* iteration to obtain bounded inversion for nonlinear NMP systems. The essential features of this method are briefly introduced in the following paragraphs.

This approach can be divided into three steps: firstly, obtain the zero dynamics by local coordinate transformation. The zero dynamics of a nonlinear system is the internal dynamics of the system subject to the constraint that the outputs, and therefore all the derivatives of the outputs, are set to zero for all time. The second step involves checking whether the approximation of the zero dynamics satisfies a certain condition that guarantees that the system is locally approximately linear in a neighbourhood of the equilibrium point (*Condition* 1 in Devasia *et al* (1996)). If so, there is a theorem (*Theorem* 1 in Devasia *et al* (1996)) which assures a unique solution. Finally, based on the linearised zero dynamics, the *Picard-like* iteration is adopted to provide bounded inverse solutions. Afterwards, with the availability of the solutions from the internal system and then through inverting the transformation matrix, the bounded inputs can finally be found. The details of this approach are as follows:

If a square nonlinear system is considered, with the number of its inputs (q) equal to the number of outputs, the system can be represented by the following form:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}) \cdot \boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{2.1}$$

where $\boldsymbol{u} \in \mathbb{R}^q$ is the input vector, $\boldsymbol{y} \in \mathbb{R}^q$ is the output vector, and $\boldsymbol{x} \in \mathbb{R}^m$ is the state variable vector. The variables $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{h}$ are the function matrices with the corresponding orders. In addition, it is assumed here that the system has well-defined vector relative

degree $\boldsymbol{r} = [\boldsymbol{r}_1, \ \boldsymbol{r}_2, \ \cdots \ \boldsymbol{r}_q]^T$ and $|\boldsymbol{r}| = \boldsymbol{r}_1 + \boldsymbol{r}_2 \cdots + \boldsymbol{r}_q < m$ (see Appendix-A), and then the nonlinear system in Eq. (2.1) can be converted to a normal form by local coordinate transformation. The coordinates can be chosen in the following way:

$$\begin{aligned} \boldsymbol{\phi}_1^i(\boldsymbol{x}) &= \boldsymbol{h}_i(\boldsymbol{x}) \\ \boldsymbol{\phi}_2^i(\boldsymbol{x}) &= L_f \boldsymbol{h}_i(\boldsymbol{x}) \\ &\cdots \\ \boldsymbol{\phi}_{\boldsymbol{r}_i}^i(\boldsymbol{x}) &= L_f^{\boldsymbol{r}_i - 1} \boldsymbol{h}_i(\boldsymbol{x}) \end{aligned} \qquad (2.2)$$

$$1 \le i \le q$$

where $\boldsymbol{L}$ stands for the *Lie* derivative. Now, a new notation is introduced for the transformed system:

$$\boldsymbol{\xi}^i = \begin{bmatrix} \boldsymbol{\xi}_1^i \\ \boldsymbol{\xi}_2^i \\ \cdots \\ \boldsymbol{\xi}_{\boldsymbol{r}_i}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_1^i(\boldsymbol{x}) \\ \boldsymbol{\phi}_2^i(\boldsymbol{x}) \\ \cdots \\ \boldsymbol{\phi}_{\boldsymbol{r}_i}^i(\boldsymbol{x}) \end{bmatrix} \qquad (2.3)$$

Then, if $|\boldsymbol{r}| < m$, it is always possible to find $m - |\boldsymbol{r}|$ more functions $\boldsymbol{\phi}_{|\boldsymbol{r}|+1}(\boldsymbol{x})$, ···, $\boldsymbol{\phi}_m(\boldsymbol{x})$ such that the $\boldsymbol{\phi}(\boldsymbol{x})$ vector has a nonsingular Jacobian matrix at $\boldsymbol{x}_0$ (where $\boldsymbol{x}_0$ is an arbitrary point). Furthermore, provided the distribution

$$\boldsymbol{G} = span\{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_m\} \qquad (2.4)$$

is involutive near $\boldsymbol{x}_0$, it is always possible to calculate the $\boldsymbol{\phi}_{|\boldsymbol{r}|+1}(\boldsymbol{x})$, ···, $\boldsymbol{\phi}_m(\boldsymbol{x})$ based on the following equation:

$$L_{\boldsymbol{g}_j} \boldsymbol{\phi}_i(\boldsymbol{x}) = 0 \qquad (2.5)$$

$$|\boldsymbol{r}| + 1 \le i \le m \qquad 1 \le j \le q$$

If we let:

$$\boldsymbol{\eta}(\mathrm{t}) = \begin{bmatrix} \boldsymbol{\eta}_1(\mathrm{t}) \\ \boldsymbol{\eta}_2(\mathrm{t}) \\ \cdots \\ \boldsymbol{\eta}_{m-|\boldsymbol{r}|}(\mathrm{t}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_{|\boldsymbol{r}|+1}(x) \\ \boldsymbol{\phi}_{|\boldsymbol{r}|+2}(x) \\ \cdots \\ \boldsymbol{\phi}_m(x) \end{bmatrix} \qquad (2.6)$$

then based on the above definitions, it is possible to derive the final transformed form:

$$
\begin{aligned}
&\dot{\xi}_1^i(\mathrm{t}) = \xi_2^i(\mathrm{t}) \\
&\cdots \\
&\dot{\xi}_{r_i-1}^i(\mathrm{t}) = \xi_{r_i}^i(\mathrm{t}) \\
&\dot{\xi}_{r_i}^i(\mathrm{t}) = b_i[\xi(\mathrm{t}),\eta(\mathrm{t})] + \sum_{j=1}^{q} a_{ij}[\xi(\mathrm{t}),\eta(\mathrm{t})]u_j(\mathrm{t}) \\
&y_i(\mathrm{t}) = \xi_{r_i}^i(\mathrm{t})
\end{aligned}
\tag{2.7}
$$

where $a_{ij}[\xi(\mathrm{t}),\eta(\mathrm{t})] = L_{g_j} L_f^{r_i-1} h_i\{\phi^{-1}[\xi(\mathrm{t}),\eta(\mathrm{t})]\}$ and $b_i[\xi(\mathrm{t}),\eta(\mathrm{t})] = L_f^{r_i-1} h_i\{\phi^{-1}[\xi(\mathrm{t}),\eta(\mathrm{t})]\}$.

This also can be represented as a compact form:

$$
\boldsymbol{y}^{(r)}(t) = \boldsymbol{B}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)] + \boldsymbol{A}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)] \cdot \boldsymbol{u}(t) \tag{2.8}
$$

where

$$
\begin{aligned}
&\boldsymbol{y}(\boldsymbol{t}) = [\boldsymbol{y}_1(t), \quad \boldsymbol{y}_2(t), \quad \cdots \quad \boldsymbol{y}_q(t)]^T \\
&\boldsymbol{u}(\boldsymbol{t}) = [\boldsymbol{u}_1(t), \quad \boldsymbol{u}_2(t), \quad \cdots \quad \boldsymbol{u}_q(t)]^T
\end{aligned}
\tag{2.9}
$$

$$
\boldsymbol{B}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)] = L_f^{r-1}\boldsymbol{h}\{\boldsymbol{\phi}^{-1}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)]\} \tag{2.10}
$$

$$
\boldsymbol{A}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)] = L_g L_f^{r-1}\boldsymbol{h}\{\boldsymbol{\phi}^{-1}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)]\} \tag{2.11}
$$

and the undriven system (or the zero dynamics) is:

$$
\dot{\boldsymbol{\eta}}(t) = \boldsymbol{\alpha}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)] + \sum_{i=1}^{q} \boldsymbol{\beta}_i[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)]\boldsymbol{u}_i(t) \tag{2.12}
$$

As a results of the assumption of the full relative degree, the term $\boldsymbol{A}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)]$ is nonsingular. Thus, the input $\boldsymbol{u}(t)$ can be found from Eq. (2.8) as:

$$
\boldsymbol{u}(t) = \boldsymbol{A}^{-1}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)] \cdot \{\boldsymbol{y}^{(r)}(t) - \boldsymbol{B}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)]\} \tag{2.13}
$$

Moreover, as far as the inversion-based tracking problem is concerned, the outputs are designated to follow the desired manoeuvre $\boldsymbol{y}_d(t)$. They are also the inputs in the inverse system. Therefore, the following equation is defined:

$$
\boldsymbol{\xi}(t) = \boldsymbol{\xi}_d(t) = [\boldsymbol{\xi}_d^1(t), \quad \boldsymbol{\xi}_d^2(t), \quad \cdots \quad \boldsymbol{\xi}_d^q(t)]^T \tag{2.14}
$$

where $\boldsymbol{\xi}_d^i(t) = [\boldsymbol{y}_{id}^1(t) \quad \boldsymbol{y}_{id}^2(t) \quad \cdots \quad \boldsymbol{y}_{id}^{(r_i-1)}(t)]$ and $\boldsymbol{y}_{id}(t)$ is the i_{th} element of the desired manoeuvre $\boldsymbol{y}_d(t)$. Thus, the zero dynamics driven by the ideal manoeuvre can be shown as:

$$
\dot{\boldsymbol{\eta}}(t) = \boldsymbol{s}[\boldsymbol{\xi}_d(t),\boldsymbol{\eta}(t),\boldsymbol{y}_d^{(r)}(t)] \tag{2.15}
$$

The system represented by Eq. (2.15) is a core subsystem. The term NMP originates from here because there are right-half plane (RHP) poles in its linearised form around the equilibrium point. If this zero dynamic system satisfies *Condition* 1 in the reference (Devasia *et al*., 1996) then by *Theorem* 1 of that reference, the solution $\boldsymbol{\eta}(t)$ for Eq. (2.15) can be found by following the steps described below. Consequently it is easy to obtain the required inputs from Eq. (2.13).

Step 1: the linearization of Eq. (2.15). In Eq. (2.15), the term $\boldsymbol{\xi}_d(t)$ is constructed from $\boldsymbol{y}_d(t)$ as shown in Eq. (2.14). Thus, it should be eliminated from the equations and after linearization, the following equation is obtained:

$$\dot{\boldsymbol{\eta}}(t) = \boldsymbol{A}_{\boldsymbol{\eta}} \cdot \boldsymbol{\eta}(t) + \boldsymbol{\psi}[\boldsymbol{\eta}(t), \boldsymbol{y}_d^{(r)}(t)] \tag{2.16}$$

where $\boldsymbol{A}_{\boldsymbol{\eta}}$ is the linearised term around the origin and the nonlinear term $\boldsymbol{\psi}(\cdot)$ is defined as the following:

$$\boldsymbol{\psi}[\boldsymbol{\eta}(t), \boldsymbol{y}_d^{(r)}(t)] = \boldsymbol{s}[\boldsymbol{\xi}_d(t), \boldsymbol{\eta}(t), \boldsymbol{y}_d^{(r)}(t)] - \boldsymbol{A}_{\boldsymbol{\eta}} \cdot \boldsymbol{\eta}(t) \tag{2.17}$$

Step 2: Decouple the $\boldsymbol{A}_{\boldsymbol{\eta}}$ into stable and unstable subsystems. The results are:

$$\boldsymbol{A}_{\boldsymbol{\eta}} = \begin{bmatrix} \boldsymbol{A}_{\boldsymbol{s}} & 0 \\ 0 & \boldsymbol{A}_{\boldsymbol{u}} \end{bmatrix} \tag{2.18}$$

where the eigenvalues of $\boldsymbol{A}_{\boldsymbol{s}}$ and $\boldsymbol{A}_{\boldsymbol{u}}$ are on the left and right-hand sides of the complex plane, respectively. If there are poles on the imaginary axis, which is termed the *nonhyperbolic* problem, the method introduced in the next section has to be adopted.

Step 3: Implementation of the *Picard-like* iteration. This algorithm mainly involves solving iteratively the following two equations:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{s},N}(t) = \boldsymbol{A}_{\boldsymbol{s}} \cdot \boldsymbol{\eta}_{\boldsymbol{s},N}(t) + \boldsymbol{I}_{\boldsymbol{s}} \cdot \boldsymbol{\psi}\{[\boldsymbol{\eta}_{\boldsymbol{s},N-1}(t), \boldsymbol{\eta}_{\boldsymbol{u},N-1}(t)]^{\boldsymbol{T}}, \boldsymbol{y}_d^{(r)}(t)\} \tag{2.19}$$

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{u},N}(t) = \boldsymbol{A}_{\boldsymbol{u}} \cdot \boldsymbol{\eta}_{\boldsymbol{u},N}(t) + \boldsymbol{I}_{\boldsymbol{u}} \cdot \boldsymbol{\psi}\{[\boldsymbol{\eta}_{\boldsymbol{s},N-1}(t), \boldsymbol{\eta}_{\boldsymbol{u},N-1}(t)]^{T}, \boldsymbol{y}_d^{(r)}(t)\} \tag{2.20}$$

where the dimensions of the unit matrices $\boldsymbol{I}_{\boldsymbol{s}}$ and $\boldsymbol{I}_{\boldsymbol{u}}$ are equal to the row dimension of $\boldsymbol{A}_{\boldsymbol{s}}$ and $\boldsymbol{A}_{\boldsymbol{u}}$ respectively. Now solve Eq. (2.19) using forward integration and Eq. (2.20) using backward integration as shown in the following equations:

$$\boldsymbol{\eta}_N(t)=\begin{bmatrix}\boldsymbol{\eta}_{s,N}(t)\\ \boldsymbol{\eta}_{u,N}(t)\end{bmatrix}=\begin{bmatrix}\int_{-\infty}^{t}\Phi_s(t,\tau)\boldsymbol{I}_s\boldsymbol{\psi}_{s,N-1}(\tau)d\tau\\ -\int_{t}^{\infty}\Phi_u(t,\tau)\boldsymbol{I}_u\boldsymbol{\psi}_{u,N-1}(\tau)d\tau\end{bmatrix} \tag{2.21}$$

where

$$\Phi_s(t_1,t_2)=e^{\boldsymbol{A}_s(t_1-t_2)},\ \Phi_u(t_1,t_2)=e^{\boldsymbol{A}_u(t_1-t_2)}$$

are the transition matrices of Eqs. (2.19) and (2.20). The terms $\boldsymbol{\psi}_{s,N-1}(\tau)$ and $\boldsymbol{\psi}_{u,N-1}(\tau)$ are the nonlinear terms of Eqs. (2.19) and (2.20).

In addition to the above $\boldsymbol{\eta}_N(t)$ solution, if the internal system satisfies the hyperbolicity and its nonlinear term locally meets the Lipschitz-like condition, then the iterative solution $\boldsymbol{\eta}_N(t)$ will converge to the ideal value $\boldsymbol{\eta}(t)$. If $\boldsymbol{\eta}(t)$ is available, the input $\boldsymbol{u}(t)$ can easily be found. The introduced approach works in the time interval $-\infty<t<+\infty$. The interval $(-\infty, t]$ is for the stable part of the linearised internal system and $[t, +\infty)$ for the corresponding unstable part. Provided the linearised internal system is hyperbolic and the residual nonlinearity is Lipschitz continuous with small linear bounds, guaranteed by *Condition* 1, it is possible to find a bounded solution for the unstable internal system. Since the future-time information is utilized, this approach is a noncausal method. In addition, the solution process depends on the linearization around the equilibrium point. Thus, its valid domain is usually restricted to tracking trajectories involving small amplitudes and slow variations.

2.2.1.2 Inversion of nonhyperbolic systems

Systems which have zeros on the imaginary axis are called nonhyperbolic systems. The aforementioned approach fails to apply to this kind of situation due to the infinite pre-actuation time for general output trajectories. The solution for this case has been provided by (Devasia, 1997; 1999). The methodology there shows that this problem can be eliminated by modifying the internal system by adding an extra perturbation term $\boldsymbol{\nu}(t)$ to Eq. (2.13). This will move the zeros slightly off the imaginary axis.

$$\boldsymbol{u}(t)=\boldsymbol{A}^{-1}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)]\{\boldsymbol{y}^{(r)}(t)-\boldsymbol{B}[\boldsymbol{\xi}(t),\boldsymbol{\eta}(t)]+\boldsymbol{\nu}(t)\} \tag{2.22}$$

Here $\boldsymbol{\nu}(t)$ has the following feedback form:

$$\boldsymbol{v}(t) = \boldsymbol{F} \cdot \begin{bmatrix} \boldsymbol{e}_{\xi}(t) \\ \boldsymbol{\eta}(t) \end{bmatrix} \tag{2.23}$$

where $\boldsymbol{e}_{\zeta}(t)$ is the tracking-error variable vector. $\boldsymbol{F}$ is chosen so that the modified internal system is hyperbolic. Then the above-mentioned *Picard-like* iterative algorithm is implemented to find the solution for the internal system. This is at the cost of precision to achieve the stable inversion of the nonlinear NMP system with the nonhyperbolic internal system.

2.2.2 Causal inversion of nonlinear system dynamics

In the stable inversion approach, for the NMP system, the inversion results depend on the whole desired output trajectory. This limitation restricts its application to trajectory planning problems or to real time applications without a predetermined trajectory. To eliminate this issue, several approaches have been developed and published. Two of them will be introduced in this section. The first is termed the *causal inversion* of the NMP system (Wang & Chen, 2001; 2002b) and the second is the *preview-based* stable-inversion (Zou & Devasia, 1999; 2007). Strictly speaking, the latter cannot be categorized within the causal inversion group. However, in this thesis, based on the sense of eliminating the requirement of a whole pre-specified desired-output manoeuvre, this approach is considered as a causal inversion method.

2.2.2.1 Casual inversion of nonminimum-phase systems

The causal inversion of NMP systems is introduced first. This approach was proposed by Wang and Chen (2001) to overcome the noncausal problem of the traditional stable inversion approaches and did not require pre-actuation. They defined two causal problems: the causal inversion problem and the optimal causal inversion problem, which will be explained in the text that follows. Later the same authors slightly modified this method and designed a controller with the H_{∞} algorithm (Zhou, Doyle & Glover, 1996) to achieve stable ε-tracking of a desired manoeuvre via a causal inversion approach (Wang & Chen, 2002a; 2002b). It has been successfully applied to a one-link flexible manipulator system.

Consider the same form of a nonlinear system expressed in Eq. (2.1) and assume $\boldsymbol{f}(\mathbf{0}) = \mathbf{0}$ and $\boldsymbol{h}(\mathbf{0}) = \mathbf{0}$. The causal inversion problem can be explained as finding a nominal control input $\boldsymbol{u}_d(t)$ and a desired state trajectory $\boldsymbol{x}_d(t)$ for a smooth desired manoeuvre $\boldsymbol{y}_d(t)$, which is zero when $t \leq 0$. It also needs to meet the following three requirements:

1.) The stability of $\boldsymbol{u}_d(t)$ and $\boldsymbol{x}_d(t)$. This means these variables are bounded, and

$$\boldsymbol{u}_d(t) \to \mathbf{0}\,,\ \boldsymbol{x}_d(t) \to \mathbf{0} \ \text{ as } \ t \to \infty$$

2.) Exact output matching.

$$\boldsymbol{h}[\boldsymbol{x}_d(t)] = \boldsymbol{y}_d(t)$$

3.) The causality of $\boldsymbol{u}_d(t)$ and $\boldsymbol{x}_d(t)$. That is:

$$\boldsymbol{u}_d(t) = \mathbf{0}\,,\ \ \boldsymbol{x}_d(t) = \mathbf{0} \ \text{ when } \ t \leq 0$$

As far as the optimal causal inversion problem is concerned, the following performance index has to be defined:

$$\boldsymbol{J}[\boldsymbol{u}_d(t), \boldsymbol{x}_d(t)] = \frac{1}{2}\int_0^{\infty} \left\| \dot{\boldsymbol{x}}_d(t) - \{\boldsymbol{f}[\boldsymbol{x}_d(t)] + \boldsymbol{g}[\boldsymbol{x}_d(t)] \cdot \boldsymbol{u}_d(t)\} \right\|_{\boldsymbol{R}}^2 \qquad (2.24)$$

where $\boldsymbol{R}$ is a weighting operator.

The methodology on which the technique is built is quite similar to the aforementioned approach for eliminating the nonhyperbolic problem. This method likewise requires the local coordinate transformation to obtain Eqs. (2.8) and (2.12). After linearising the right-hand side of Eq. (2.15) at the equilibrium point $\boldsymbol{\eta} = \mathbf{0}$ and then splitting the linearised matrix into stable and unstable parts, the following two equations can be obtained:

$$\dot{\boldsymbol{\eta}}_s(t) = \boldsymbol{A}_s\boldsymbol{\eta}_s(t) + \boldsymbol{B}_s\boldsymbol{y}_d^{(r)}(t) + \boldsymbol{d}_s[\boldsymbol{\xi}_d(t), \boldsymbol{y}_d^{(r)}(t), \boldsymbol{\eta}_s(t), \boldsymbol{\eta}_u(t)] \qquad (2.25)$$

$$\dot{\boldsymbol{\eta}}_u(t) = \boldsymbol{A}_u\boldsymbol{\eta}_u(t) + \boldsymbol{B}_u\boldsymbol{y}_d^{(r)}(t) + \boldsymbol{d}_u[\boldsymbol{\xi}_d(t), \boldsymbol{y}_d^{(r)}(t), \boldsymbol{\eta}_s(t), \boldsymbol{\eta}_u(t)] \qquad (2.26)$$

where the matrices $\boldsymbol{A}_s$ and $\boldsymbol{A}_u$ are, respectively, the stable and unstable parts of the linearised matrix, and $\boldsymbol{d}_s(\cdot)$ and $\boldsymbol{d}_u(\cdot)$ represent the higher-order terms of the expressions. Now if an appropriate additional term $\boldsymbol{\upsilon}(t)$ can be selected to add into Eq. (2.26), the unstable system can reach a condition of asymptotic stability. The following two equations represent the two resulting systems:

$$\dot{\bar{\boldsymbol{\eta}}}_s(t) = \boldsymbol{A}_s\bar{\boldsymbol{\eta}}_s(t) + \boldsymbol{B}_s\boldsymbol{y}_d^{(r)}(t) + \boldsymbol{d}_s[\boldsymbol{\xi}_d(t), \boldsymbol{y}_d^{(r)}(t), \bar{\boldsymbol{\eta}}_s(t), \bar{\boldsymbol{\eta}}_u(t)] \qquad (2.27)$$

$$\dot{\bar{\eta}}_u(t) = A_u\bar{\eta}_s(t) + B_u y_d^{(r)}(t) + d_u[\xi_d(t), y_d^{(r)}(t), \bar{\eta}_s(t), \bar{\eta}_u(t)] + \upsilon(t) \tag{2.28}$$

If the equation $\upsilon(t) = K\bar{\eta}_u(t) - d_u[\xi_d(t), y_d^{(r)}(t), \bar{\eta}_s(t), \bar{\eta}_u(t)]$ is selected, Eq. (2.28) becomes:

$$\dot{\bar{\eta}}_u = (A_u + K)\bar{\eta}_u(t) + B_u y_d^{(r)} \tag{2.29}$$

where K is chosen to make (A_u+K) *Hurwitz*. Thus, the stable solution $\bar{\eta}_u(t)$ can be obtained from Eq. (2.29) and then $\bar{\eta}_s(t)$ from Eq. (2.27). Finally according to Eq. (2.13), the bounded input $\bar{u}_d(t)$ can be found.

If the performance index in Eq. (2.17) is adopted to facilitate the selection of the $\upsilon(t)$, it has been shown that the optimal causal inversion has similarities with the *Minimum Energy Control Problem* (MECP) which is often met in the control field. Based on this point, for a linear system, the $\upsilon(t)$ can be successfully found. Wang and Chen (2001) also pointed out that the above method can be used to find an optimal noncausal solution if the whole trajectory of a smooth desired manoeuvre $y_d(t)$ is given.

This method has shows some strong points over the nonlinear regulation method and the noncausal method for NMP systems. Unlike the nonlinear regulation approach, it can avoid the numerical intractability of solving nonlinear partial differential equations (PDE) and the combined transient errors are significantly smaller. In addition, for the linear system application, MECP has shown that in the above process of derivation it is only necessary to consider the unstable internal system.

2.2.2.2 Preview-based stable inversion

This method was firstly proposed by Zou and Devasia (1999) for linear systems. The purpose of this proposed approach is to eliminate one of the defects existing in the traditional stable inversion methods for NMP systems. Later, Zou and Devasia (2007) extended its application to nonlinear NMP systems. It makes use of the finite-time-window $[t_c, t_c+T_p]$ preview information of the desired output instead of the whole future information $[t_c, +\infty)$. In fact, for practical situations, sometimes one has to implement this preview-based stable inversion method due to the finite range of sensors in a vehicle or the fact that

the future desired trajectory needs to be updated online etc. One of the major contributions of this paper is that it presents the quantification method to calculate the required T_p in terms of the desired accuracy of the calculated inputs and the internal dynamics. In addition, Zou and Devasia pointed out that the required preview time T_p has a close relationship with the smallest real part of the unstable poles of the linearised internal system. This method is introduced in this section.

A finite desired trajectory $\boldsymbol{y}_d(t)$ in the time widow $[t_c, t_c+T_p]$ is used to compute the bounded solution for Eq. (2.21). Here a boundary condition is assumed:

$$\boldsymbol{B}_o = \begin{bmatrix} \boldsymbol{\eta}_s(t_c) \\ \boldsymbol{\eta}_u(t_f) \end{bmatrix} \tag{2.30}$$

where $t_f = t_c+T_p$. The *Picard-like* algorithm is adopted here. Thus, the initial values for this algorithm can be chosen as:

$$\boldsymbol{\eta}_0(t) = \begin{bmatrix} \Phi_s(t,t_c)\boldsymbol{\eta}_s(t_c) \\ \Phi_u(t,t_f)\boldsymbol{\eta}_u(t_f) \end{bmatrix} \qquad \text{for } t\in[t_c, t_c+T_p] \tag{2.31}$$

Then for the iteration number $N>1$:

$$\boldsymbol{\eta}_N(t) = \begin{bmatrix} \boldsymbol{\eta}_{s,N}(t) \\ \boldsymbol{\eta}_{u,N}(t) \end{bmatrix} = \begin{bmatrix} \Phi_s(t,t_c)\boldsymbol{\eta}_s(t_c) + \int_{-\infty}^{t} \Phi_s(t,\boldsymbol{\tau})\boldsymbol{I}_s\boldsymbol{\psi}_{s,N-1}(\boldsymbol{\tau})d\boldsymbol{\tau} \\ \Phi_u(t,t_f)\boldsymbol{\eta}_u(t_f) - \int_{t}^{t_f} \Phi_u(t,\boldsymbol{\tau})\boldsymbol{I}_u\boldsymbol{\psi}_{u,N-1}(\boldsymbol{\tau})d\boldsymbol{\tau} \end{bmatrix} \tag{2.32}$$

When N tends to infinity, the solutions of Eq. (2.32) are the same as those obtained from Eq. (2.21). However, the above method requires the future boundary conditions $\boldsymbol{\eta}_u(t_f)$, which are usually unknown a priori. The paper further pointed out that this quantity can be set to zero for the boundary condition matrix. The error in computing the internal dynamics under this new boundary condition is bounded and the error also decays exponentially as the preview time T_p increases. After $\boldsymbol{\eta}_u(t_f)$ is available, all other values can be computed.

2.2.3 Developments in terms of other inversion techniques

Other authors have also contributed significantly to the development of model inversion techniques. In their paper Hunt and Meyer (1997) improved the method of Devasia *et al*

(1996) by presenting a two-step noncausal procedure. They stated that the latter method in fact is applied to an error system instead of to the full system. The first step of their method computes the major part of the desired control and the corresponding state trajectory by ignoring the perturbation error. The second step finds the required control and states by computing a noncausal and stable solution to an error-driven dynamical equation.

In all the above-mentioned methods, it is assumed the nonlinear system represented in Eq. (2.1) has strictly relative degree. In practice, this assumption sometimes can be violated. Ramakrishna *et al* (2001) investigated how the relative degree of the system can change with respect to parameter values. Their results show that although the order of the differential equation can change (the relative degree changes), the inversion solution is still continuous at the nominal value point of the varying parameter.

2.3 Review of traditional inverse simulation algorithms

In this section, the interest is focused upon a review of the available techniques for inverse simulation.

2.3.1 Classification of inverse simulation approaches

Inverse simulation aims to determine the system inputs required to produce a given response, defined in terms of the system output variables. Interest in inverse simulation methods has been particularly strong in the field of aircraft flight mechanics and this approach has received special attention in the case of helicopters and other forms of rotorcraft, which involve complex and highly nonlinear models. For such an application the input to the inverse simulation is the required flight path and the output information represents the piloting commands needed to achieve this trajectory.

Inverse simulation is commonly carried out either by a direct approach based on differentiation or iteratively using integration methods. The first published accounts of the

problems of inverse simulation for aircraft applications were those of Kato and Saguira (1986) and Thomson (1987). Their methods involved numerical differentiation of the vehicle state variables, with respect to time. The main advantage of this approach is fast convergence speed. However, it may suffer from problems of numerical rounding error and involves ad-hoc approaches for specific applications for different types of vehicles. Sentoh and Bryson (1992) defined the inverse process as a LQ optimal problem that minimises the integral of a weighted square sum of the deviations from a straight flight path and control surface deflections. They demonstrated the approach by an application to feedforward control problems in an aeronautical context. However, this method suffered from significant practical limitations and involved a relatively cumbersome procedure.

In the early 1990s members of a research group at the University of California, Davis (Gao & Hess, 1993; Hess *et al*., 1991) proposed what is now the most commonly used approach that formulates the inverse problem as part of an integration process. This approach does not require time differentiation of the specified path constraints. Instead, it involves a procedure that calculates the partial derivative of the output vector with respect to the input vector through a numerical algorithm. In addition, redundancy problems can be overcome by use of the Moore-Penrose inverse. Unlike the earlier methods based on the differentiation approach, the structure of this algorithm determines that this integration-based method is less model-specific. That means that it can accommodate different models without restructuring the algorithm itself. One of the drawbacks of this approach is that it is an order of magnitude slower than the method involving differentiation with respect to time.

In an approach similar to the integration-based algorithm, de Matteis *et al* (1995) presented an alternative local optimisation concept to eliminate the control redundancy problem. This involved adding new path constraints at the cost of evaluating the Hessian matrix numerically though a modified *Broyden, Fletcher, Goldfar and Shanno* (BFGS) quasi-Newton method. However, in practice, it is not always feasible to construct new path constraints for a special performance requirement as this approach may not always lead to a solution due to the searching region being restricted. By incorporating a two timescale approach to simplify the complexity of aircraft models, this method has been successfully

demonstrated on a F-16 fighter aircraft model (Avanzini *et al*., 1999) and a Bell aH-1G single rotor helicopter model (Avanzini & de Matteis, 2001).

Lee and Kim (1997) formulated the inverse simulation problem as a general optimisation problem by defining a performance index constrained by equality conditions that are a function of state variables. Then the performance index is discretized by the finite element method and the final governing equation is solved by the Levenberg-Marquardt (LM) algorithm. This can avoid the control redundancy problem by appropriate selection of the performance index and constraint condition. Therefore, the procedure does not involve numerical differentiation or integration processes. As a result, it overcomes the problem of ill conditioning and sensitivity problems associated with initial guessed values. However, the performance improvement is achieved at the cost of enormously increased complexity of the inverse simulation process.

Celi (2000) solved the inverse problem by borrowing some ideas from the optimisation field. In fact, unlike the approaches based on local optimisation, his method considers pilot inputs as design variables in the global space. Furthermore, investigations based on indirect trajectory definition are allowed and as a result, the vehicle dynamics obtained (in Celi's case a helicopter) and the required pilot inputs are noticeably different due to the existence of a family of valid trajectories. However, the approach may have difficulty in calculating the whole trajectory at one time. If this is the case, the results show poor consistency between the converged solution and the desired trajectory. This problem can be solved by performing the optimisation over overlapping consecutive segments of the trajectory rather than over the entire trajectory. In addition, problems of multiple solutions may appear. This will assist in handling qualities studies but creates difficulties if the inverse solution is being used for purposes of simulation model validation. As a consequence, additional constraints are required to achieve a unique solution.

Finally, other authors also have made contributions to the inverse simulation field. Anderson (2003) proposed an enhanced NR method by combining Hess's approach with the bisection method through which each change of controls is multiplied by an additional

scale factor. He stated that the numerical stability of the inverse simulation can be significantly improved and an order of magnitude reduction can be achieved in both the tracking error and the control deflections, as shown from results obtained from an application to a helicopter model with an individual blade representation. In fact, his method can be considered as another modification of the calculation of the Jacobian matrix and is quite similar to the inverse Broyden method (Cheney & Kincaid, 2004), but simpler. However the bisection method has some drawbacks. One disadvantage is that when the searching interval for a real root is decreased, the speed of convergence becomes very slow due to the computational load. It is also difficult to achieve high accuracy using this approach.

Lu *et al* (2007c) proposed an approach based on sensitivity-analysis (SA) to solve some numerical problems existing in the traditional integration method. The details will be presented in the next chapter. In addition, Lu, Murray-Smith and McGookin (2007a) developed a derivative-free approach to improve the numerical stability. This allowed considering the inclusion of the actuator saturation in the model being investigated as well as discontinuous manoeuvres. This approach will also be discussed in later sections of this thesis.

Based on the above literature review, the various techniques may be categorised as follows:

Methods in which derivative information is used:

- ***Optimisation methods***

  Local optimisation:

  The LQ problem (Sentoh & Bryson, 1992); the local optimisation approach with the BFGS algorithm (de Matteis *et al*., 1995); the two timescale approach (Avanzini *et al*., 1999; Avanzini & de Matteis, 2001); the method based on sensitivity analysis (Lu *et al*., 2007c).

  Global optimisation:

  The general optimisation problem involving equality conditions (Lee & Kim, 1997); the optimisation approach of Celi (2000).

- ***Differentiation methods***

  Numerical differentiation of the vehicle constrained variables, with respect to time, until the control variables are solved explicitly (Kato & Saguira, 1986; Thomson, 1987; Thomson & Bradley, 1997).

- ***Integration methods***

  The value of the control variables that satisfy the constraints are found iteratively within a sampling interval (Gao & Hess, 1993; Hess *et al*., 1991; Rutherford & Thomson, 1996); modification of the Hess approach by the bisection method (Anderson, 2003).

Methods that do not involve derivative information:

- ***Optimisation methods***

  A derivative free approach based on the NM algorithm is used to find the control values within a sampling interval (developed in the course of the current research: details in a later section (Lu *et al*., 2007a).

2.3.2 The differentiation-based approach

Because it is completely different from the most widely adopted integration-based approaches and because it has been used successfully in a variety of previous applications, the algorithm of the differentiation-based approach is reviewed in this section (Murray-Smith, 2000).

A nonlinear system, whose form is slightly different from the one used for the traditional model inversion as shown in Eq. (2.1), may be described by equations of the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.33}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.34}$$

where $\boldsymbol{f} \in \mathbb{R}^m$ is the set of nonlinear ordinary differential equations describing the original system, $\boldsymbol{g} \in \mathbb{R}^p$ is the set of algebraic equations that construct the expected outputs, and $\boldsymbol{u} \in \mathbb{R}^q$ is the input vector. $\boldsymbol{x} \in \mathbb{R}^m$ is the state-variable vector and $\boldsymbol{y} \in \mathbb{R}^p$ is the vector of output variables. This form follows the traditional definition used in the previous inverse

simulation investigation (Hess *et al*., 1991). In addition, Eqs. (2.33) and (2.34) can be discretized as:

$$\frac{\boldsymbol{x}(t_k)-\boldsymbol{x}(t_{k-1})}{t_k-t_{k-1}}=\boldsymbol{f}[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k)] \qquad k=1,\,2,\,3,\ldots,\,N-1 \tag{2.35}$$

$$\boldsymbol{y}(t_k)=\boldsymbol{g}[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k)] \tag{2.36}$$

where N is the total number of discretized intervals and t_k is the k_{th} discretization point in the time period. Now define two functions $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ to calculate out the values of the unknown variables $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$.

$$\boldsymbol{F}_1[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k)]=\boldsymbol{f}[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k)]-\frac{\boldsymbol{x}(t_k)-\boldsymbol{x}(t_{k-1})}{t_{k-1}-t_k} \tag{2.37}$$

$$\boldsymbol{F}_2[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k)]=\boldsymbol{g}[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k)]-\boldsymbol{y}_d(t_k) \tag{2.38}$$

where the term on the left-hand side of Eq. (2.36) is replaced by ideal output values $\boldsymbol{y}_d(t_k+1)$, and where the subscript d is used to represent the desired value. The NR method is adopted to solve Eq. (2.37) and Eq. (2.38) so that the values $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$ can make the right hand sides of these equations approximately equal to zero. The updated equations are shown as follows:

$$\begin{bmatrix}\boldsymbol{x}^{(n)}(t_k)\\ \boldsymbol{u}^{(n)}(t_k)\end{bmatrix}=\begin{bmatrix}\boldsymbol{x}^{(n\text{-}1)}(t_k)\\ \boldsymbol{u}^{(n\text{-}1)}(t_k)\end{bmatrix}-\begin{bmatrix}\dfrac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{u}}\\ \dfrac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{x}}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{F}_1[\boldsymbol{x}^{(n\text{-}1)}(t_k),\boldsymbol{u}^{(n\text{-}1)}(t_k)]\\ \boldsymbol{F}_2[\boldsymbol{x}^{(n\text{-}1)}(t_k),\boldsymbol{u}^{(n\text{-}1)}(t_k)]\end{bmatrix} \tag{2.39}$$

where the quantity n is the current step within the iterative process. After the values $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$ that make $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ zero are found, the inverse simulation will move to the next time step t_{k+1}. By similar sequential steps, the complete time histories of $\boldsymbol{x}(t_k)$ and $\boldsymbol{u}(t_k)$ can eventually be obtained.

2.3.3 The integration-based approach

This section provides a summary of the integration-based method of Hess *et al* (1991). After the discretization process of Eqs. (2.33) and (2.34), the input-output relationship of this nonlinear system can be defined as follows:

$$\boldsymbol{x}(t_{k+1}) = \int_{t_k}^{t_{k+1}} \dot{\boldsymbol{x}}(t)dt + \boldsymbol{x}(t_k) \tag{2.40}$$

$$\boldsymbol{y}(t_{k+1}) = \boldsymbol{g}[\boldsymbol{x}(t_{k+1}), \boldsymbol{u}(t_k)] \tag{2.41}$$

Now the term on the left-hand side of Eq. (2.41) is replaced by ideal output values $\boldsymbol{y}_d(t_{k+1})$, where the subscript d is used to represent the desired value. Thus, Eq. (2.41) can be rewritten as:

$$\boldsymbol{g}[\boldsymbol{x}(t_{k+1}), \boldsymbol{u}(t_k)] - \boldsymbol{y}_d(t_{k+1}) = 0 \tag{2.42}$$

In the traditional algorithm, the NR-based method is used to find $\boldsymbol{u}(t_k)$ by the following iterative relationship:

$$\boldsymbol{u}^{(n+1)}(t_k) = \boldsymbol{u}^{(n)}(t_k) - (\boldsymbol{J}\{\boldsymbol{f}_E[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)]\})^{-1} \boldsymbol{f}_E[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)] \tag{2.43}$$

where $\boldsymbol{f}_E[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)] = \boldsymbol{g}[\boldsymbol{x}^{(n)}(t_{k+1}), \boldsymbol{u}^{(n)}(t_k)] - \boldsymbol{y}_d(t_{k+1})$. In Eq. (2.43) the term $\boldsymbol{J}$ represents the Jacobian matrix of system outputs at the end of the time interval Δt (from t_k to t_{k+1}) with respect to input variables. If, in Eq. (2.34), there is a direct analytical relationship between input and output the Jacobian matrix may be obtained directly. Otherwise an approximation technique must be used as follows:

$$\boldsymbol{J}_{ij} = \frac{\partial \boldsymbol{y}_i(t_{k+1})}{\partial \boldsymbol{u}_j(t_k)} \approx \frac{\partial \boldsymbol{y}_i(\boldsymbol{u}_j + \Delta \boldsymbol{u}_j)|_{t_{k+1}} - \partial \boldsymbol{y}_i(\boldsymbol{u}_j)|_{t_{k+1}}}{\Delta \boldsymbol{u}_j} \tag{2.44}$$

$$i = 1, 2, 3, \ldots, p \text{ and } j = 1, 2, 3, \ldots, q$$

where $\boldsymbol{u}_j$ and $\boldsymbol{y}_i$ are the j_{th} and i_{th} elements of the input and output vectors, respectively. $\Delta \boldsymbol{u}_j$ is the perturbation in $\boldsymbol{u}_j$ at time t_k. In Eq. (2.44), the superscript n is omitted. Rutherford and Thomson (1996) have also presented a modified approach for calculation of the Jacobian matrix by perturbing $\Delta \boldsymbol{u}_j$ in the negative and positive directions.

When a redundant situation exists, the Jacobian matrix is not square and it is not possible to use standard methods of matrix inversion in the NR iteration scheme, as shown in Eq. (2.43). Hess *et al* (1991) proposed the use of the pseudo-inverse matrix as a solution for finding the roots of Eq. (2.42) when $\boldsymbol{J}$ is rectangular.

2.4 Manoeuvre definition

From the definition of inverse simulation, it is clear that this technique can be implemented to calculate the control efforts required by a modelled vehicle in following a specified manoeuvre. Therefore, for all cases, the initial stage of any inverse simulation must involve construction of the trajectory or manoeuvre that is to be investigated. In the helicopter field the applications of inverse simulation have been concerned particularly with aggressive flying tasks involving ADS-33E mission task elements (MTE) (Anon, 2000). Hence, the manoeuvre definition involves problems associated with the conversion of implicit words describing MTE in the ADS-33E manual into precise flight paths.

Many contributions have been made to this translation process for the special case of helicopter flight mechanics (Rutherford & Thomson, 1997; Thomson & Bradley, 1990a; 1994). A manoeuvre is usually composed of two parts: the positional coordinates $[x_e(t) \quad y_e(t) \quad z_e(t)]$ and the heading angle Ψ. The former elements are used to describe the motion of the helicopter's centre of gravity relative to an Earth-fixed system of axes. The inclusion of the latter (heading angle) aims to provide the fourth constraint to produce a unique solution as well as to describe a realistic manoeuvre and vehicle movement, since there are four controls in the helicopter dynamic system. In addition, the choice of the heading angle from the vehicle attitude angles is convenient in terms of flight mechanics. By following this definition, this subsection will present mathematical models for the pop-up, hurdle-hop, and bob-up manoeuvres. Another approach (Avanzini *et al*., 1998; Avanzini and de Matteis, 2001), which is different from the method adopted here, can also be found quite useful to construct an ideal trajectory. This approach implemented the Frenet triad to construct the algebraic equations to describe the trajectory.

2.4.1 The pop-up manoeuvre

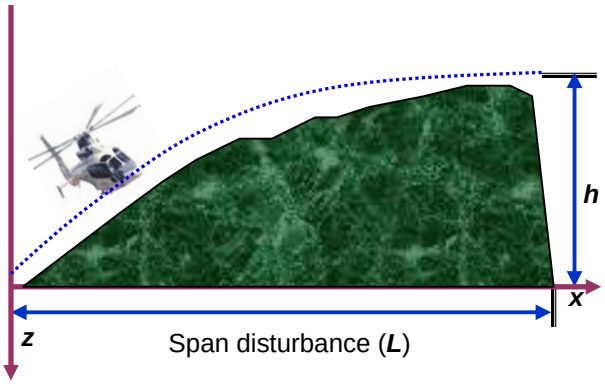

Fig. 2.2 The pop-up manoeuvre

In Fig. 2.2, h is the height of the obstacle and L is the total track distance. As shown, the pop-up manoeuvre involves clearing an obstacle by a rapid controlled change of altitude over some distance L in the x - z vertical plane. The trajectory shape shown in the above figure can be described by a polynomial equation with some boundary conditions, following the specifications required in the ADS-33E manual (Rutherford & Thomson, 1997; Thomson & Bradley, 1998). A fifth-order polynomial has been found to satisfy these boundary conditions and to guarantee the necessary smoothness of the manoeuvre (although sometimes higher order polynomials may be needed). This polynomial has the form:

$$z_e(t) = -[6(\frac{t}{t_m})^5 - 15(\frac{t}{t_m})^4 + 10(\frac{t}{t_m})^3]h \qquad 0 < t < t_m \tag{2.45}$$

where t_m is the time to complete the manoeuvre and. In practical applications, the total flight speed V_f can be either constant (which is usually the case) or a variable as a function of time. The horizontal velocity component can be derived as follows:

$$\dot{x}_e(t) = \sqrt{V_f(t)^2 - \dot{z}_e(t)^2} \tag{2.46}$$

The quantity t_m for the hurdle-hop manoeuvre is calculated by means of the following equation:

$$L = \int_0^{t_m} \dot{x}_e(t)dt \tag{2.47}$$

where x_e is the longitudinal displacement. Eq. (2.47) can be easily solved by forward iteration. In practical applications, the values of the parameters h, V_f and L can be adjusted to provide the desired level of severity for the manoeuvre.

2.4.2 The hurdle-hop manoeuvre

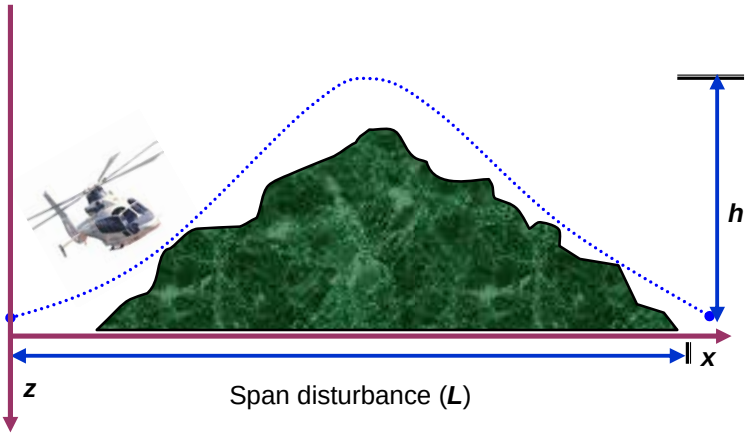

Fig. 2.3 The hurdle-hop manoeuvre

As shown in Fig. 2.3, the hurdle-hop manoeuvre involves clearing an obstacle of height h and then returning back to the original altitude over some span distance L (Rutherford & Thomson, 1996). The apex of the manoeuvre occurs in the middle of the span. Similar to the approach used for constructing the pop-up manoeuvre, the polynomial equation used to describe the hurdle-hop manoeuvre can be defined as follows:

$$z_e(t) = 64h[(\frac{t}{t_m})^3 - 3(\frac{t}{t_m})^2 + 3(\frac{t}{t_m}) - 1](\frac{t}{t_m})^3 \tag{2.48}$$

where the value of the parameter t_m can be calculated in a similar fashion to Eq. (2.47).

2.4.3 The bob-up manoeuvre

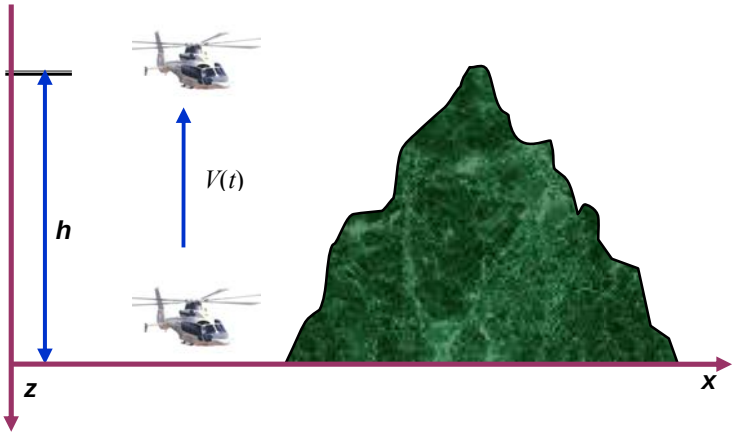

Fig. 2.4 The bob-up manoeuvre

The bob-up manoeuvre, as shown in Fig. 2.4, is started from a trimmed state (hover) and then power is increased to increase the vertical velocity to approximately maximum at the middle time point of the manoeuvre. Then, power is decreased to reduce the vertical velocity which finally reaches zero at the end point (Anon, 2000; Bradley & Thomson, 1993). The velocity description of this process can be illustrated in Fig. 2.5:

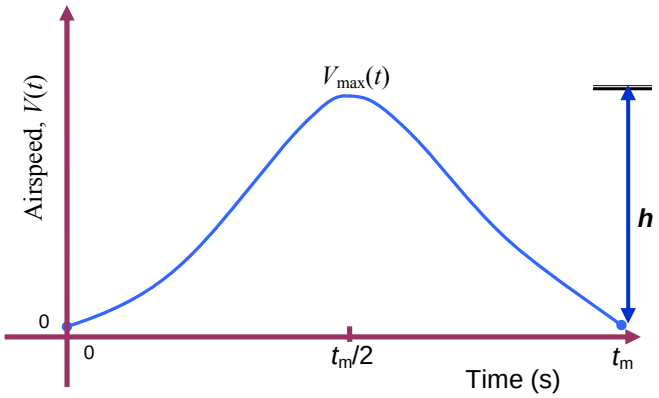

Fig. 2.5 Velocity profile of the bob-up manoeuvre

The polynomial equation used to describe the bob-up manoeuvre can be defined as follows:

$$V(t) = V_{\max}[-64(\frac{t}{t_m})^6 + 192(\frac{t}{t_m})^5 - 192(\frac{t}{t_m})^4 + 64(\frac{t}{t_m})^3] \tag{2.49}$$

where V_{max} is the maximum velocity in the manoeuvre.

2.5 Summary

In this chapter historic developments in the field of dynamic inversion and inverse simulation have been summarised. Both classes of approaches can be used to derive the inputs for a given trajectory but they rely on entirely different methodologies. Model inversion based methods obtain the input by inverting a nonlinear dynamic system model in advance. In contrast, inverse simulation methods are used to find the input based on special numerical algorithms or optimisation approaches. In addition, compared with feasibility of the principles of inverse simulation, the currently available techniques for dynamic inversion are complex and tedious to apply, especially for the methods attempting to achieve the noncausal inversions for NMP systems. This information has served to drive the current research to explore the possibility of replacing model inversion by inverse simulation for the synthesis of FFCs within the 2DOF control scheme.

Chapter 3

The Relationship between Model Inversion and Inverse Simulation

Contents

The primary objective of this chapter is to explore and highlight the close relationship between model inversion and inverse simulation techniques. The similarities and shortcomings, existing in these two classes of methods, are presented. All these findings are intended to facilitate investigation of the possibility of replacing model inversion by inverse simulation in the design of a FFC. The work presented in this chapter also has been published in the *Proc. of 5th MATHMOD* (Lu, Murray-Smith & McGookin, 2006b) and *Journal of Mathematical and Computer Modelling of Dynamical Systems* (Lu *et al*., 2007a).

3.1 Introduction

As mentioned in Chapter 1 and Chapter 2, inversion of system dynamics is a widely investigated approach used to design the FFC to obtain precision tracking of output trajectories through a combination of feedforward and state feedback controllers (Devasia *et al*., 1996). Compared with MP systems, the tracking problem for a NMP system is much

more difficult since limitations introduced by unstable zero dynamics are structural and cannot be avoided without changing the system structure or reformulating the tracking problem. In addition, these traditional model inversion methods require extra efforts to overcome some drawbacks such as the complexity of the algorithm structures, the demand for sufficient smoothness of the manoeuvre and model being investigated, and the limited domain of validity etc. All these problems provide the stimulus for the development of new methodologies that share the same functionality as the model inversion techniques but are less complicated to apply and more feasible to implement in practical situations.

Chapter 1 has introduced the idea that inverse simulation can be used in a similar way to model inversion techniques in control system design. In their pioneering work Sentoh and Bryson (1992) developed an approach to realize feedforward command generators for a guidance controller by solving an inverse problem. Hess and Gao (1993) formulated task-driven bandwidth requirements for the design of a stability augmentation system using an inverse solution. Gray and von Grünhagen (1998) use a 2DOF control structure in which the FFC channel is replaced by a direct pilot input as an approximation to an inverse simulation. This structure can facilitate investigating the quality of the developed mathematical model as well as possible sources of the inaccuracy. A later contribution made by Boyle and Chamitoff (1999) involved an application for an autonomous guidance system for an unmanned aerial vehicle (UAV). Meanwhile, Avanzini *et al* (1999) combined the inverse problem and a (LQ) tracking controller to achieve good tracking performance and robust stability. The inverse approach is used to determine the input commands necessary to track the specified flight path and the aspect of the design process concerned with robust stability takes account of unmodelled dynamics and external disturbances. This idea of using a combination of approaches was further presented in the work of Avanzini (2004) where he explored the possibility of inverse simulation being used to provide the reference input for a controlled helicopter model. There he first simplified the original model using a two time-scale approach and then applied traditional inverse simulation methods.

These examples of previous work show that inversion simulation shares some common features with model inversion. They both need a way to generate the ideal manoeuvre and then use different mathematical approaches to provide the reference inputs for the controlled model from the defined manoeuvre. These similarities provide some sound reasons to replace the inverse model by inverse simulation, if the latter can satisfy all the necessary conditions. However, earlier investigations of inverse simulation methods have not fully considered the applicability of this approach for the special case of NMP systems. Yip and Leng (1998) failed to make a clear statement about why inverse simulation can be applied successfully to the NMP problem. Moreover, their development was based the assumption of fast convergence of the NR method. However, this assumption may be not always valid, as is shown later in this chapter.

3.2 The approach of Yip and Leng (1998)

In order to establish a basis for comparison with the approach presented in this chapter, the main content of Yip and Leng's method is reviewed in this subsection. Their analysis is based on the integration-based method for inverse simulation (Hess *et al*., 1991). Firstly, the Jacobian matrix in Eq. (2.44) is assumed to be constant in the iteration process for a linear time-invariant system. Accordingly, Eq. (2.43) will be changed to the following form:

$$\begin{aligned}
&\boldsymbol{n}=0, \quad \boldsymbol{u}^{(1)}(t_k)=\boldsymbol{u}^{(0)}(t_k)+\boldsymbol{J}^{-1}\boldsymbol{e}^{(0)}(t_k)\\
&\boldsymbol{n}=1, \quad \boldsymbol{u}^{(2)}(t_k)=\boldsymbol{u}^{(0)}(t_k)+\boldsymbol{J}^{-1}[\boldsymbol{e}^{(0)}(t_k)+\boldsymbol{e}^{(1)}(t_k)]\\
&\qquad\qquad \vdots\\
&\boldsymbol{n}=l-1, \ \boldsymbol{u}^{(l)}(t_k)=\boldsymbol{u}^{(0)}(t_k)+\boldsymbol{J}^{-1}[\boldsymbol{e}^{(0)}(t_k)+\boldsymbol{e}^{(1)}(t_k)+\cdots+\boldsymbol{e}^{(l-1)}(t_k)]
\end{aligned} \tag{3.1}$$

where l is the total number of iterations and $\boldsymbol{e}(t_k)=-\boldsymbol{f}_{\boldsymbol{E}}(t_k)$. A multiplier, ω, then is added to improve the initial guess such that

$$\boldsymbol{u}^{(l)}(t_k)=\boldsymbol{u}^{(0)}(t_k)+\boldsymbol{J}^{-1}\omega\boldsymbol{e}^{(0)}(t_k) \tag{3.2}$$

Compared with Eq. (3.1), the following equation can be obtained:

$$\omega\boldsymbol{e}^{(0)}(t_k)=\boldsymbol{e}^{(0)}(t_k)+\boldsymbol{e}^{(1)}(t_k)+\cdots+\boldsymbol{e}^{(l-1)}(t_k) \tag{3.3}$$

Now if good convergence of the inverse simulation process is assumed within the first little iteration for a small Δt value, the value ω can be set approximately to unity. In addition, the term $\boldsymbol{u}^{(0)}(t_k)$ is replaced by $\boldsymbol{u}(t_{k-1})$. Therefore, after omitting the superscript *n*, Eq. (3.2) becomes

$$\boldsymbol{u}(t_k) = \boldsymbol{u}(t_{k-1}) + \boldsymbol{J}^{-1}\boldsymbol{e}(t_k) \tag{3.4}$$

where $\boldsymbol{e}(t_k) = \boldsymbol{y}_d(t_k) - \boldsymbol{y}(t_k)$. Eq. (3.4) is termed an approximate model of Newton's scheme. With the simplified relationship among the input, the Jacobian matrix, and the error function, Yip and Leng investigated stability of the inverse simulation process for an aircraft application.

3.3 The essence of inverse simulation

Analysis of the inverse simulation process for the case of a nonlinear system of the type shown in Eq. (2.1) is difficult. Therefore, it is more appropriate to first consider a linear system having the form shown in Eq. (3.5).

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{aligned} \tag{3.5}$$

where $\boldsymbol{x}$ is the vector of system state variables, $\boldsymbol{u}$ is the input vector, $\boldsymbol{y}$ is the output vector, and the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are the system matrices with the appropriate dimensions.

The inverse simulation procedure based on the integration process may be divided into two stages: first the discretization process and then the solution by means of the numerical algorithms, as shown Eq. (2.43). This division is reasonable because many other inverse simulation methodologies also follow this kind of two-stage structure, using other numerical algorithms instead of the NR approach (Avanzini & de Matteis, 2001; de Matteis *et al*., 1995; Lee & Kim, 1997; Lu *et al*., 2007c). The stability of the second stage usually relates to the numerical stability and convergence properties of the chosen algorithm itself. This involves numerical issues more than questions of dynamical stability. As a result, only

the discussion of the first stage is presented and for the second stage convergence is assumed to be achievable.

After discretizing Eq. (3.5), the following formulae can be obtained:

$$\begin{aligned} \boldsymbol{x}(t_{k+1}) &= \boldsymbol{P}\boldsymbol{x}(t_k) + \boldsymbol{H}\boldsymbol{u}(t_k) \\ \boldsymbol{y}(t_k) &= \boldsymbol{C}\boldsymbol{x}(t_k) + \boldsymbol{D}\boldsymbol{u}(t_k) \end{aligned} \tag{3.6}$$

where the terms $\boldsymbol{P}$ and $\boldsymbol{H}$ are:

$$\begin{aligned} \boldsymbol{P} &= e^{\boldsymbol{A}\Delta t} \\ \boldsymbol{H} &= \left(\int_0^{\Delta t} e^{\boldsymbol{A}t}\boldsymbol{d}t\right)\boldsymbol{B} \end{aligned} \tag{3.7}$$

For the inverse simulation method introduced in Chapter 2, the state variables are first updated using Eq. (2.40) using the fourth-order Runge-Kutta (RK) algorithm. If the RK algorithm is applied for the integration process of the right side of Eq. (3.7), Eq. (2.40) can be expressed by the following equation after transformation and simplification:

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{Q}(\boldsymbol{A}, M, \Delta t)\boldsymbol{x}(t_k) + \boldsymbol{W}(\boldsymbol{A}, \boldsymbol{B}, M, \Delta t)\boldsymbol{u}(t_k) \tag{3.8}$$

where the variable M is the number of iterative RK steps for one integration step from t_k to t_{k+1}. The function $\boldsymbol{Q}$ is dependent on the algebraic relationship of the three variables $\boldsymbol{A}$, M, and Δt. The function $\boldsymbol{W}$ also depends on the matrix $\boldsymbol{B}$ in addition to the three other quantities shown. When the value M is increased, the accuracy of the results from Eq. (3.8) will be improved at the cost of greatly increased complexity. If M tends to infinity, Eq. (3.8) will be identical to Eq. (3.6). It can thus be concluded that the inverse simulation approximates to the process of discretization and the accuracy of this approximation depends on the value of M. In addition, the zeros of the system in Eq. (3.6) can be relocated in the z-plane by varying the sampling rate Δt. In the practical inverse simulation process, the values $\boldsymbol{A}$, $\boldsymbol{B}$, and M in Eq. (3.8) are usually fixed. Hence, by changing the value Δt in Eq. (3.8), it may be possible to redistribute the zeros in the z-plane as in Eq. (3.6) and to avoid the NMP problem.

The application of the new method to the NMP problem can be explained as follows. Assume first that the system shown in Eq. (3.6) is a NMP system and has RHP zeros, regardless of the distribution of poles. This process of disregarding the poles is possible

because only the RHP zeros will affect the dynamic stability of the inverse system. As mentioned above, by changing the value of Δt it is possible to move zeros originally in the RHP into the left-half plane (LHP). This can guarantee the stability of the inverse simulation process in terms of the system structure at the first stage. Hence, there may exist some sampling-rate intervals or critical Δt_c values where the magnitudes of all the zeros are less than one.

Moreover, even if some magnitudes are greater than unity, inverse simulation may still provide good convergence because of the fact that it approximates to but is not exactly the same as a traditional discretization process. However, it is difficult to obtain Δt_c directly from Eq. (3.8) due to the complicated structures of the two functions $\boldsymbol{Q}$ and $\boldsymbol{W}$. Furthermore, this complexity is greatly increased when the value of M is increased. In practical terms, Δt_c can be obtained from Eq. (3.6) by plotting a diagram showing the distribution of magnitudes of zeros versus the sampling-rate variation. These values of Δt_c can then be taken as the reference Δt values for Eq. (3.8). As M tends to infinity, values obtained from Eq. (3.6) should be quite close to those obtained from Eq. (3.8).

The analysis presented in this section is different from that given by Yip and Leng (1998). Firstly, they addressed the stability analysis using an assumption of fast convergence of the NR method, typically within two or three steps, as shown in Section 3.2, instead of the two-stage division. However, this assumption of fast convergence may not be appropriate for cases where the inverse simulation does converge but at a relatively slow rate. This situation is quite usual for many cases, even for a very simple case that will be illustrated later. Moreover, their assumption is made for the case of small Δt values. However, it is well known that small Δt values will lead to some numerical instabilities such as the high-frequency oscillations discussed previously (Lin, 1993). Secondly, in practice, the assumption of the constant Jacobian matrix, or the existence of the direct analytic relationship between input and out, may not be satisfied for many situations (Gao & Hess, 1993; Hess *et al*., 1991). This assumption of small Δt values can be avoided entirely in the new approach described here. Thirdly, the two methods are based on different standpoints in terms of investigation of the stability of the inverse simulation process. The Yip and

Leng approach mainly focused on the approximation of the NR algorithm. In contrast, the approach presented in this section is concerned more with the first stage – the discretization process. Finally, Yip and Leng failed to make a clear statement about why inverse simulation can be applied successfully to the NMP problem.

Although there are some shortcomings in Yip and Leng's method, it can work well provided all of the above assumptions are satisfied. Compared with their method, the two-stage methodology introduced in this chapter is more general and can work for a range of different situations. Taken overall the essential feature of inverse simulation based on the integration process for a linear system is that it approximates the original system by using a discrete equivalent. The same analysis can be applied to the nonlinear case but the procedure is more complex and challenging since it involves discretization of a nonlinear system.

3.4 Application examples

The proposed approach will be illustrated at length in this subsection through three different applications to allow a more detailed description and demonstration of the methodology. The three cases to be considered are: a nonlinear MP case, a linear NMP case, and a *multiple-input multiple-output* (MIMO) NMP case.

3.4.1 A nonlinear minimum-phase system

The simulation study selected here relates to a nonlinear longitudinal mathematical model of a fixed-wing aircraft, the HS125 (Hawker 800) business jet (Thomson, 2004) (Appendix-B). It can be shown that the linearised model for this aircraft around the chosen equilibrium point is a MP system since there are no RHP zeros for this model. The thrust T (N) and the elevator angle δ_e (deg) act as the inputs for implementation of the algorithm for inverse simulation involving the NR approach. The manoeuvre conducted is a constant forward-speed hurdle-hop manoeuvre (Rutherford & Thomson, 1996) in the z-x plane (altitude

versus distance travelled), as introduced in Eq. (2.48) in Chapter 2. It may be characterized by the following polynomials:

$$Z_d(t) = 64h\left[(\frac{t}{t_m})^3 - 3(\frac{t}{t_m})^2 + 3(\frac{t}{t_m}) - 1\right](\frac{t}{t_m})^3 \text{m}$$
$$V_{fd}(t) = 61.87 \text{m} \cdot \text{s}^{-1} \tag{3.9}$$

where t_m is the time to complete the manoeuvre and can be calculated in a similar fashion to Eq. (2.47), h is the height. Eq. (3.9) also shows that the total flight speed V_f remains constant during the manoeuvre.

In this application the first priority is to define the calculated manoeuvre based on the vector relative degree (Appendix-A), if it exists. Calculations show that the model has a vector relative degree [2, 1]. This means that the manoeuvre must be defined in terms of acceleration [see Eq. (2.14)] for application of the model inversion approach. To guarantee a fair comparison, the ideal manoeuvre is also defined as the acceleration in the inverse simulation, although it is not essential in this case. This is one of the advantages of implementation of inverse simulation to derive the require inputs.

The methods applied are based on completely different fundamental methodologies for deriving the inputs. Inverse simulation obtains the inputs and reference states one by one in each fixed time interval Δt and it does not necessarily require trajectory derivative information. In contrast to inverse simulation, model inversion techniques calculate the inverse model in advance and then carry out the forward simulation using the defined manoeuvre with the corresponding derivatives of appropriate order. In addition, the traditional model inversion approaches require the original system to be of full relative degree and involves a complex local-coordinate transformation (Sastry, 1999). The simulation results are generated for the conditions defined in Eq. (3.10) and are shown in Fig. 3.1 and Fig. 3.2:

$$h = 150 \text{ m}; \quad s = 500 \text{ m} \tag{3.10}$$

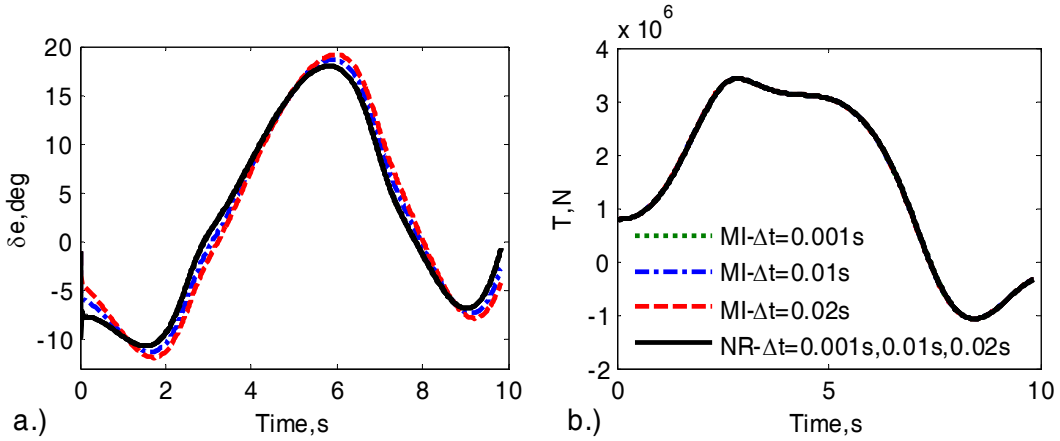

Fig. 3.1 Inputs from inverse simulation (NR) and model inversion (MI) for the HS125 model

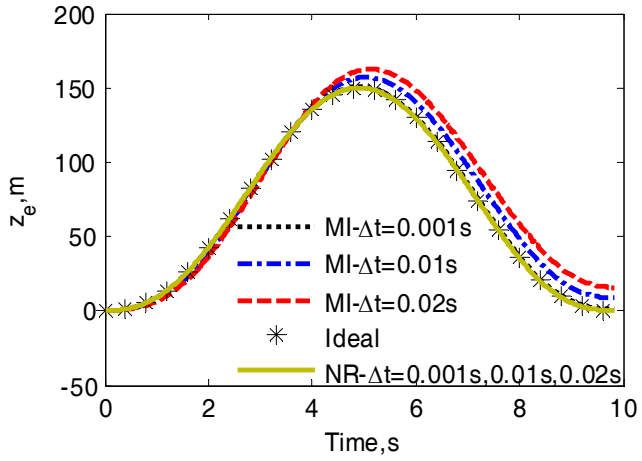

Fig. 3.2 Comparisons of outputs from forward simulation for the ideal manoeuvre for the HS125 model

Fig. 3.1 and Fig. 3.2 show that for this case inverse simulation shows more accurate results compared with the model inversion for the larger Δt values such as 0.01 s and 0.02 s. In Fig. 3.1b, both methods obtain the same thrust (T) for all Δt values being investigated. However, for the elevator angle (δ_e) channel (Fig. 3.1a), there are differences between the results for Δt =0.01 s and Δt =0.02 s. The results from the forward simulation with these calculated inputs, as shown in Fig. 3.2, further illustrates the poor consistency of the model inversion for Δt =0.01 s and Δt =0.02 s. The latter only achieves good results for Δt =0.001 s. However, use of this smaller Δt value means increased computation time.

In addition to the more accuracy for this case, compared with the model inversion techniques, inverse simulation is easier and more feasible in terms of implementation. Properties of the algorithm mean that the original system does not require the existence of the vector relative degree. It is therefore suggested that for MP systems, particularly for applications where the model is quite complex, such as in a helicopter or ship model, it would be more convenient to adopt inverse simulation, by selecting a suitable sampling interval, rather than model inversion. The chosen Δt value should satisfy two important conditions: a.) to guarantee the convergence of the inverse simulation process; b.) to ensure that the zeros in Eq. (3.8) remain in the LHP in the discretization process. These two requirements follow the property of the two-stage-division analysis, as already mentioned. The latter requirement must be included because inverse simulation approximates to the discretization process.

3.4.2 A linear SISO nonminimum-phase system

The main objective of this subsection is to illustrate the application of the methodology introduced in Section 3.2 and to show the weakness of the assumption of fast convergence of the inverse simulation process, made by Yip and Leng (1998) in the development of their method.

Consider a linear SISO NMP system given by the following four system matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ -7 \\ 81 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad D = [0] \tag{3.11}$$

This system has two RHP zeros: $0.5000 \pm 7.0534i$. Obviously, the method of Devasia *et al* (1996), as presented in Chapter 2, can be applied to overcome this NMP problem but it quite tedious and is also a noncausal process. Instead, for implementation of the method developed above, a plot of the magnitude of the zeros versus Δt values is created, as shown in Fig. 3.3.

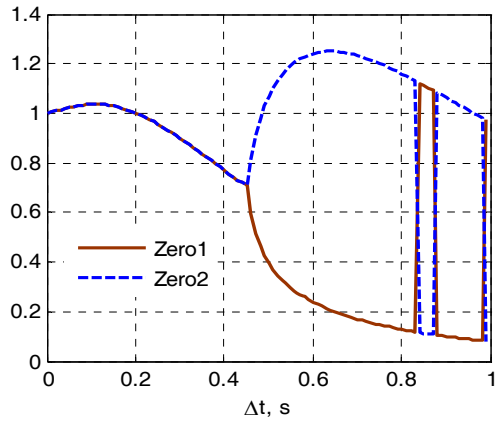

Fig. 3.3 Variation of magnitude of the zeros with Δ*t*

The interval, in which the magnitudes of the zeros are smaller than one, can be determined directly from examination of Fig. 3.3. In this case the interval is [0.2 s, 0.46 s]. For the interval [0, 0.2 s], there are clearly zeros with magnitude slightly larger than one and for the range above 0.46 s magnitudes again become greater than unit as the interval increases. The range of intervals that should be considered first for Δ*t* in the inverse simulation algorithm is therefore [0.2 s, 0.46 s]. However, it should be noted that the interval [0, 0.2 s] may not necessarily be invalid and a trial and error process may be used to check whether or not it is usable. According to the analysis presented above, the inverse simulation approximates to but is not exactly the same as a traditional discretization process and the interval [0, 0.46 s] therefore may be considered for the process of Δ*t* value selection. Simulation results support the fact that the point 0.46 s is a critical limit for convergence of the inverse simulation. However, in addition to the reasons relating to Fig. 3.3, convergence problems may also be linked to numerical limitations of the NR method implemented in the inverse simulation algorithm. Therefore, the critical point value 0.46 s is a coincidence of the combination effects from the discretization process and the NR algorithm. The results with Δ*t* values in the selected interval are shown in Fig. 3.4 for the hurdle-hop manoeuvre of Eq. (3.9).

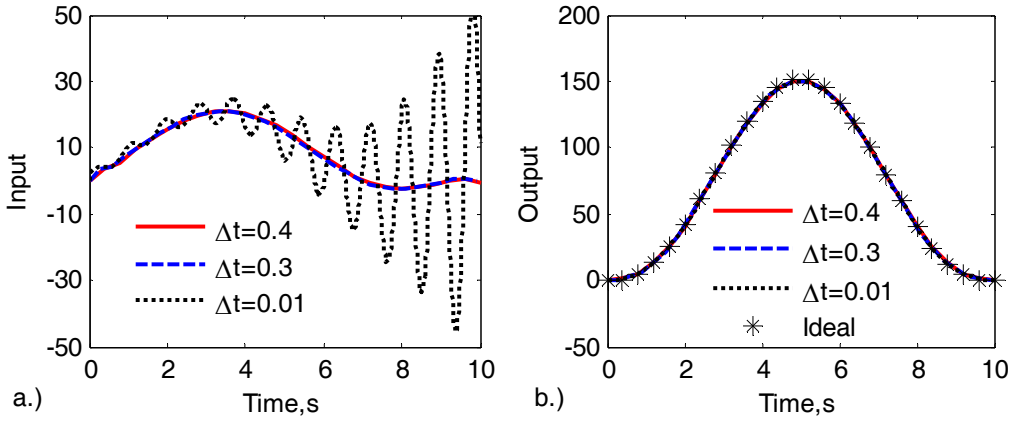

Fig. 3.4 Comparisons of results from inverse simulation with the different Δ*t* values

Fig. 3.4a shows that for the sampling rates 0.4 s and 0.3 s, the inverse simulation achieved perfectly bounded inputs. However, for Δt = 0.01 s the calculated input is combined with slowly increasing oscillations. This is consistent with the above analysis that states that the results obtained for values outside the interval [0.2 s, 0.46 s] are of lower quality and even invalid compared with Δt = 0.4 s and Δt = 0.3 s. Furthermore, the poorer results of this case conflict with the traditional idea that smaller Δt values in discretization lead to more accurate results (Hess *et al*., 1991).

An interesting phenomenon shown in Fig. 3.4b is that the results from the forward simulation with the three different calculated inputs completely satisfy the requirements for the ideal trajectory. This actually shows a multi-solution phenomenon with regard to the selection of the different Δt values for a NMP system. Therefore, special attention should be paid to deal with the selection of a suitable Δt value associated with NMP systems since this is a special case. All in all, this example demonstrates the validity of earlier statements concerning the application of inverse simulation to NMP systems.

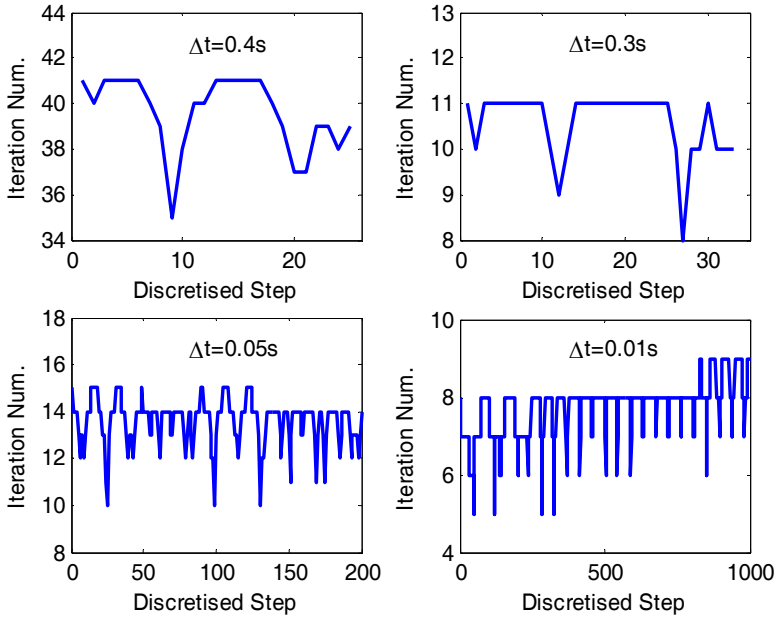

Fig. 3.5 Iterations required in inverse simulation for each discretized step

It should be noted that the method of Yip and Leng (1998) may fail because the assumption of high-speed convergence to derive Eq. (3.4) will be violated. This is shown in Fig. 3.5 where the number of iterations required for four different Δt values are shown as a function of the discretized step number. In that figure, for Δt =0.4 s the inverse simulation requires at least thirty-five iterative steps to converge at each discretized time step. Even with the smallest value Δt =0.01 s, which is located outside idea range of intervals and leads to an unbounded solution, nine iterations are needed to converge at each step. Therefore, for this case, stability analysis of the inverse simulation may be inaccurate if the terms ($n > 2$) are ignored in Eq. (3.3) even if the value Δt is selected from the valid interval [0.2 s, 0.46 s].

3.4.3 A linear MIMO nonminimum-phase system

In this example, a helicopter model is implemented in terms of an eighth-order description representative of a combat helicopter similar to the Westland Lynx, linearised around the hover situation. The inputs are the four basic control channels (i.e. main rotor collective

pitch (θ_0), main rotor longitudinal cyclic pitch (θ_{ls}), lateral cyclic pitch (θ_{lc}), and tail rotor collective pitch (θ_{tr})), as illustrated in Fig. 3.6.

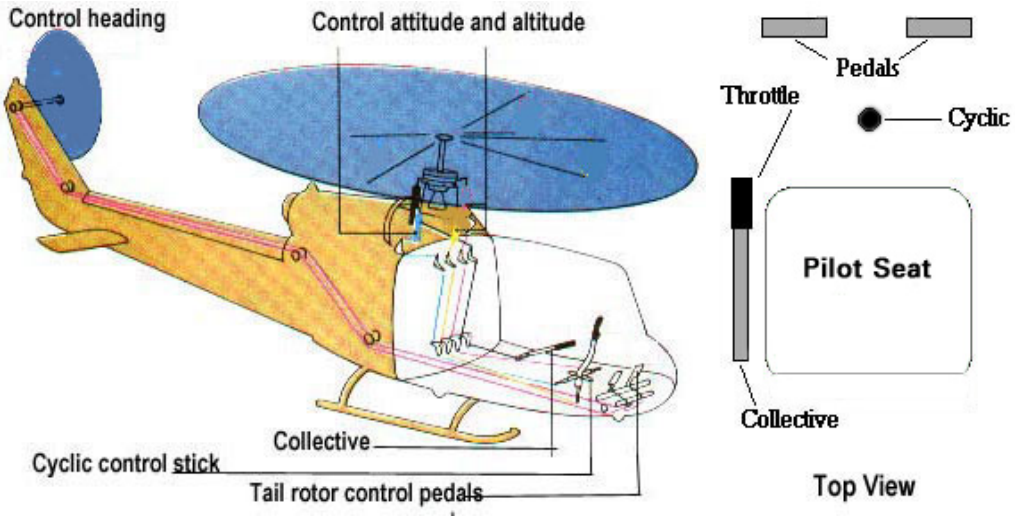

Fig. 3.6 Helicopter control system illustration (Anon, 2007)

The model has the standard state space form as shown in Eq. (3.5). Its state variable vector $\boldsymbol{x}$ contains the following system variables (Skogestad & Postlethwaite, 1996):

Table 3.1 State variables for the Westland Lynx linearised helicopter model

State Variables	Description	Unit
θ	Pitch attitude	*rad*
Φ	Roll attitude	*rad*
p	Roll rate	$rad \cdot s^{-1}$
q	Pitch rate	$rad \cdot s^{-1}$
r	Yaw rate	$rad \cdot s^{-1}$
u	Forward velocity	$ft \cdot s^{-1}$
v	Lateral velocity	$ft \cdot s^{-1}$
w	Vertical velocity	$ft \cdot s^{-1}$

The four channels of heave velocity ($\dot{H}$), roll rate (p), pitch rate (q), and heading rate ($\dot{\Psi}$) are selected to be the outputs. The desired manoeuvres of these four channels are taken from the standard heave axis response (Walker & Postlethwaite, 1996) and redefined based on the latest version of the US Army helicopter handling qualities requirements ADS-33E-PRF (Anon, 2000). The desired vertical rate response is thus defined as having the

qualitative appearance of a first-order lag with an additional pure delay, as shown in Eq. (3.12). The other three channels p, q, and $\dot{\Psi}$ are set to be zero in terms of their desired responses.

$$\dot{H}(s) = \frac{10}{0.8225 \cdot s + 1} e^{-0.162 \cdot s} \tag{3.12}$$

It can be shown easily that this Lynx-like model, for the chosen flight condition, has a vector relative degree $\boldsymbol{r} = [1 \quad 1 \quad 1 \quad 1]$. Thus, the inverse simulation is carried out using the first-order derivative of the variables for each channel for the chosen manoeuvre to get a more accurate Jacobian matrix by avoiding the traditional approximation method. The reasons for this kind of the first-order calculation will be discussed in detail in Chapter 4. The calculation to determine the zeros of the model have shown that the system has two RHP zeros and therefore is a NMP system.

As previously explained, the first step is to plot the magnitude of the zeros in the z-plane versus the sampling rate Δt. After being discretized, the system has more than two RHP zeros. The results in terms of the magnitude plot are shown in Fig. 3.7.

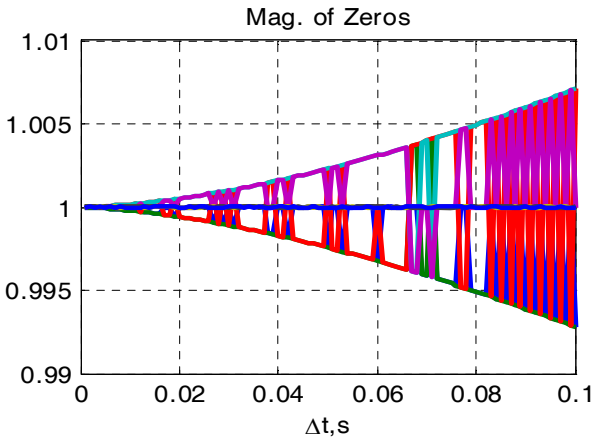

Fig. 3.7 Magnitude variation of zeros with respect to the sampling interval Δt

Fig. 3.7 shows the distribution of the magnitudes of the eight zeros of the discretized system of the Westland Lynx-like linearised model. The figure of eight zeros is determined from a series of discretization processes within the interval [0, 0.1 s]. In addition, it can be seen from Fig. 3.7 that there always exist zeros whose magnitudes are larger than unity.

Moreover, when the sampling time is increased, the magnitudes of the RHP zeros become larger. According to the previous suggestion, this means that to assure the convergence of the inverse simulation, small sampling intervals are preferred. Besides, the convergence of the NR algorithm needs to be taken into consideration. The final simulations have shown that inverse simulation can achieve convergence only when the Δt value is less than 0.05 s. Thus $\Delta t = 0.01\ s$ is selected to ensure satisfaction of the combined requirements of accuracy, numerical stability, and good convergence. The results from the simulation experiments based on this choice of Δt are shown in Figs 3.8 and 3.9.

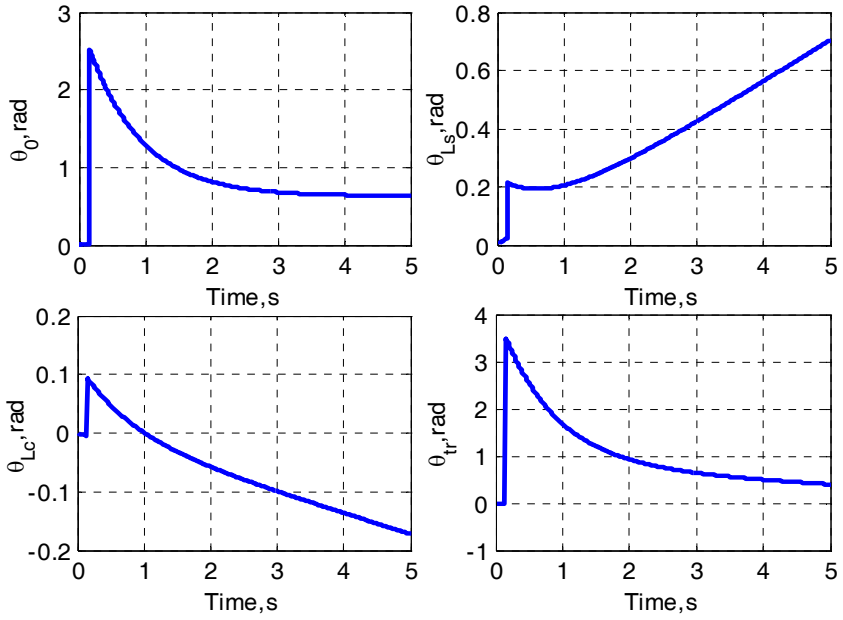

Fig. 3.8 The calculated inputs from inverse simulation (Δt = 0.01 s)

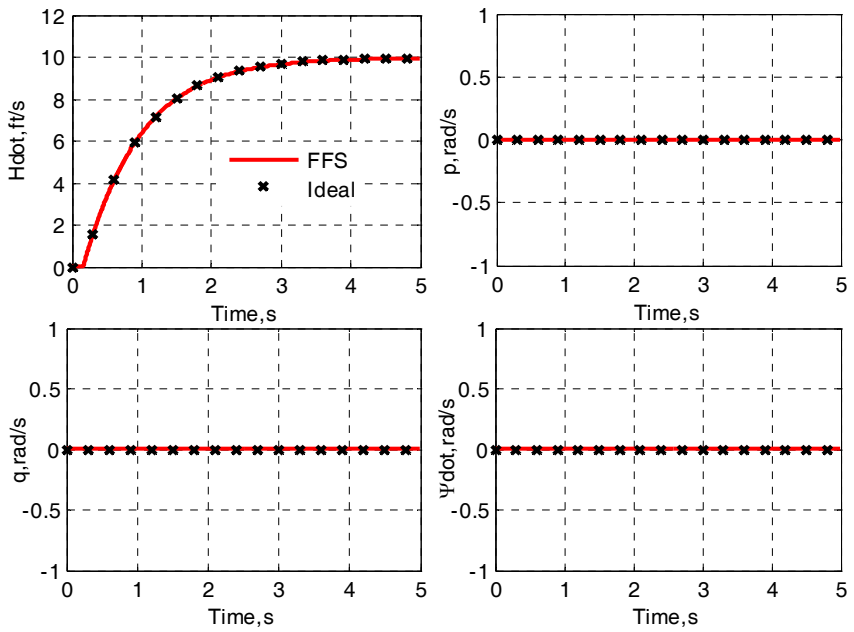

Fig. 3.9 Comparisons of the calculated outputs with the ideal manoeuvres (Δt = 0.01 s)

Fig. 3.8 shows the inputs obtained from the inverse simulation. The amplitudes of these inputs are quite large and may not have physical meaning for the linear system which is being used as a benchmark. The main rotor collective pitch (θ_0) first initiates a step input to start a heave acceleration and then decreases to a steady value after a while in order to maintain the required heave velocity. Meanwhile, a step input in the tail rotor collective pitch (θ_{tr}) is applied to balance the main rotor effect. Coupling effects can also be observed in the main rotor longitudinal (θ_{ls}) channel and the lateral cyclic pitch (θ_{lc}) channel to make the roll and pitch angles as small as possible. Fig. 3.9 shows good consistency between the ideal manoeuvres and results obtained from the forward simulation using those calculated inputs. The heave velocity ($\dot{H}$) follows the required step response while the other three channels involving roll rate (p), pitch rate (q), and heading rate ($\dot{\Psi}$) are kept at zero.

These figures show that the inverse simulation can obtain perfect results regardless of the fact that the original system (for Δt =0.01s) has three RHP zeros with magnitudes very close to one. This is consistent with that fact, mentioned previously, that the inverse simulation process can be linked to the traditional discretization process. The latter process

provides an analytical method for selecting a sub-optimal sampling interval or a reference interval for the inverse simulation. Trial and error may be involved in this process. Other tests have also been done and the results show that the inverse simulation algorithm can converge well, in this example, for values of Δt below 0.05 s. Beyond this critical point, the inverse simulation algorithm cannot converge.

When a suitable Δt value is selected, the process that has to be followed to find the reference feedforward inputs is completely causal and may be suitable for online implementation. This can bring up some advantages for situations when only partial knowledge of system state variables is available due to the limitation of the sensors or where the future desired trajectory needs to be undated online. An example of such updating of the trajectory could occur in applications such as the space-shuttle, entry-trajectory-tracking problem (Zou & Devasia, 2007) where the changes in environmental conditions are very significant. Hence, the noncausal approach (Devasia *et al*., 1996) and the more complex causal approaches (Wang & Chen, 2001; 2002b), as reviewed in Chapter 2, can be avoided. This represents one of the major advantages of inverse simulation over these other approaches.

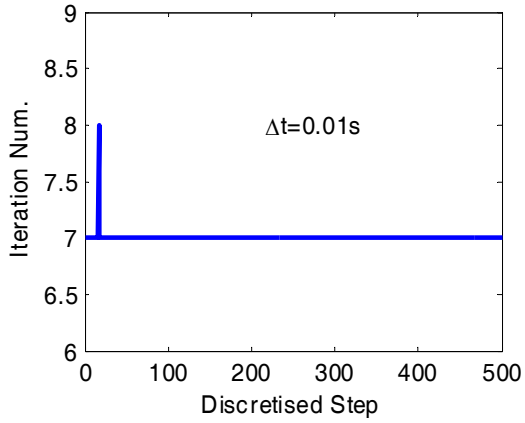

Fig. 3.10 Iterations required in inverse simulation for each discretized step

In addition, the iterative steps required during the inverse simulation process are plotted in Fig. 3.10 and this shows that the inverse simulation process needs at least seven steps for

the NR algorithm to achieve convergence in the interval $[t_k, t_{k+1}]$. This again shows the problem inherent in the assumption of fast convergence made by Yip and Leng to derive their methodology.

3.5 Summary

In this chapter, the close relationship between inverse simulation and model inversion has been explored and presented. It has been shown that it is possible and practical for inverse simulation to replace model inversion in the output-tracking field or other corresponding domains. The investigations have been carried out both on MP and NMP systems. For a suitable discretization interval, for the case of MP systems, inverse simulation can provide results that are almost identical to those obtained by model inversion. This is illustrated by an application involving a nonlinear HS125 fixed-wing aircraft model. For linear NMP systems, inverse simulation can be used successfully for causal calculation of feedforward inputs. In addition, compared with model inversion, the inverse simulation process is easier and more feasible in terms of practical implementation. This development depends upon zero redistribution within the process of inverse simulation and provides a link between the linear inverse system and its discrete counterpart in a mathematical sense. This has been successfully proved with an example of an eighth-order linear Lynx-like helicopter model. However, the investigation of inverse simulation for the case of nonlinear NMP systems requires further consideration.

In addition, compared with the Yip and Leng's method, the two-stage approach presented here is more general and less restricted. It does not require assumptions of a constant Jacobian matrix, the fast convergence etc. Moreover, the development standpoints on which these two methods depend are completely different. The Yip and Leng method focuses on the approximation of the NR algorithm while the approach presented in this chapter is based upon the approximation to the discretization process and as a result the stability of the whole inverse simulation process is affected both by the discretization process and the NR algorithm.

Chapter 4

Stability of Inverse Simulation

Contents

As well as the advantages that arise from the use of inverse simulation methods, there are potential problems of a numerical nature that can arise in solving the equations of the dynamic model in an inverse manner. This chapter focuses on investigation of phenomena involving high-frequency oscillations, redundancy problems, and the so-called "constraint oscillations" that can be encountered in applying the processes of inversion. In addition, an alternative method for calculation of the Jacobian matrix is presented. Some contents in this chapter have been contained in the paper for *Simulation Modelling Practice and Theory* (Lu, Murray-Smith, & Thomson, 2007d).

4.1 The problems of high-frequency oscillations and redundancy

It is not a trivial task to obtain the inverse response from the equations of motion of a vehicle or other system and many problems can be encountered. Because of the widespread adoption of the integration-based approach to inverse simulation (Gao & Hess, 1993; Hess *et al*., 1991) and the history of relatively successful applications with this type of method, this section focuses on numerical issues associated with the integration-based approach.

It has been commonly accepted that the algorithms can provide good results by an appropriate choice of step tolerance limit and sampling rate with appropriate initial guess values for the NR algorithm. However, as shown in Eq. (2.44), the calculation of the Jacobian matrix depends on dynamic states and its accuracy is determined by the length of the discretization interval. As a consequence, accuracy and stability problems may arise during this evaluation process. In addition, previous investigations have shown that this technique often involves redundant situations as well as failure of the inverse simulation process to converge and poor convergence or convergence with superimposed high-frequency oscillations.

The phenomenon of the high-frequency oscillations in inverse simulation solutions has been extensively investigated (Hess *et al.,* 1991; Lin, 1993; Rutherford & Thomson, 1996; 1997). Hess *et al* (1991) described the phenomenon as involving multiple local-minima of the error equation $\boldsymbol{f}_E$ as shown in Eq. (2.43). They suggested that two modifications to the algorithm could alleviate this problem of numerical instability. The first of these involves adjusting the discretization interval; and the second relates to improvements in the method for calculating the Jacobian matrix to achieve better accuracy. In addition, they also demonstrated that the high-frequency oscillations could be smoothed by introducing a digital low-pass filter without degrading the values of the control inputs obtained.

In contrast, Lin (1993) considered this phenomenon in terms of unobservable motions that are excited during the inverse process. Furthermore, with respect to the first suggestion made by Hess *et al* (1991), Lin has emphasized that, instead of eliminating this problem, the use of a small time interval could excite uncontrolled state variables and this would further lead to the high-frequency oscillation effect.

Rutherford and Thomson (1996; 1997) tried to solve this problem in a different way. Their approach was aimed at finding a better qualified Jacobian matrix from physical reasoning, by taking account of the difference of perturbations in the controls in terms of the blade aerodynamic force variables and velocity variables. In practice, the perturbations have an instantaneous effect on the blade aerodynamic forces and therefore on accelerations, but not

on velocities and displacements. Changes in those latter variables happen only after a finite period of time. In addition, the two timescale method shows advantages in terms of numerical stability in that it eliminates the high-frequency oscillations by reducing the order of the model being investigated (Avanzini *et al*., 1999; Avanzini & de Matteis, 2001).

As far as the redundancy problem is concerned, this introduces a further difficulty due to the non-square nature of the Jacobian matrix. Hess *at al* (1991) suggest that this issue could be avoided by means of the pseudo inverse. However, this could not always guarantee the convergence of the solutions. Several researchers (de Matteis *et al*., 1995; Lee & Kim, 1997; Yip & Leng, 1998) have provided some feasible solutions for the redundancy problem but these methods have some drawbacks, as mentioned in Chapter 2. Examples include the difficulty in determining the necessary performance indexes. Finally, the SA method (Lu *et al*., 2007c) can deal with the redundancy problem as a natural consequence of the structure of this special algorithm. In addition, it also can solve the problem of high-frequency oscillations by increasing the integration number within a discretized interval. All information about this approach is presented in detail in Chapter 5.

4.2 The investigation of constraint-oscillation phenomena

In addition to the high-frequency oscillations and the redundancy problem, constraint oscillations sometimes appear in the results. Thomson and Bradley (1990a) showed that this phenomenon exists in both the real pilot input data as well as in data from inverse simulations. In addition, the occurrence and the properties of this oscillating mode can be predicted by analysing the system matrix of the corresponding inverse linearised system. Anderson (2003) suggested that the root cause of this phenomenon is associated with the severity of the constraints and the model order and then suggested that it may result from the excitation of internal zero dynamics. Anderson (2003) and also Thomson and Bradley (1990a) used dynamic analysis of the inverse linearised model to reach conclusions about whether an oscillation observed in an inverse simulation solution was a constraint oscillation or not. However, the system model linearised around an equilibrium point

cannot represent the whole envelope of the manoeuvring rotorcraft, especially in cases involving severe manoeuvres.

The current research has highlighted the need for additional consideration of the relationship between inverse simulation and model inversion. The detailed information about this relationship has been given in Chapter 3 in this thesis. In this section a SISO system is presented for illustrating the potential advantage of using model inversion to explain the constraint oscillations. The fact that in some of the previous work (Anderson, 2003; Thomson & Bradley, 1990) the definition of constraint oscillations was linked to the dynamic properties of the model being investigated further supports this idea. Finally, this section also presents results from investigations that involve variation of the sampling rate, the use of different manoeuvres, and different trim points, for the case of a nonlinear Lynx helicopter model (Bagiev, 2006).

4.2.1 A simple SISO system

Because inverse simulation and model inversion share the same objective in terms of calculation of the input from a defined output trajectory, as discussed in Chapter 3, this provides a possible way to analyse inverse simulation techniques using model inversion methods. The latter therefore may be used to analyse and explain the constraint-oscillation phenomenon which exists in the inverse simulation process. The illustration presented here can be considered as an extension of the work of Thomson and Bradley (1990a) and Anderson (2003), to show that this phenomenon is related to the internal zero system.

For example, consider the following SISO system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ -5 \\ 69 \end{bmatrix} u$$

$$y = x_1 \tag{4.1}$$

This system has a zero-pair distribution of its internal system as follows:

$$z_{1,2} = -0.5 \pm 7.0534i \tag{4.2}$$

According to the definition presented in Appendix-A, the relative degree of the above system is one ($\alpha = 1$). Therefore, the order of its internal system is two (Isidori, 1989; Sastry, 1999), with regard to the equation $m - |\boldsymbol{r}|$ given in Chapter 2. In addition, after transformation, the following inverse model can be obtained by following process presented in Chapter 2:

$$\begin{aligned} \dot{\xi} &= \dot{y}_d \\ \dot{\eta}_1 &= 5\eta_1 + \eta_2 - 5\dot{\xi} \\ \dot{\eta}_2 &= -80\eta_1 - 6\eta_2 - 6\xi + 69\dot{\xi} \\ u &= -\eta_1 + \dot{\xi} \end{aligned} \tag{4.3}$$

where y_d is the desired trajectory, and η_1 and η_2 are the state variables for the internal system. The simulated results using inverse simulation obtained for the case of a sinusoidal input applied to the given model [Eq. (4.1)] are shown in Fig. 4.1. Fig. 4.2 shows results obtained through simulation using the inverse model [Eq. (4.3)].

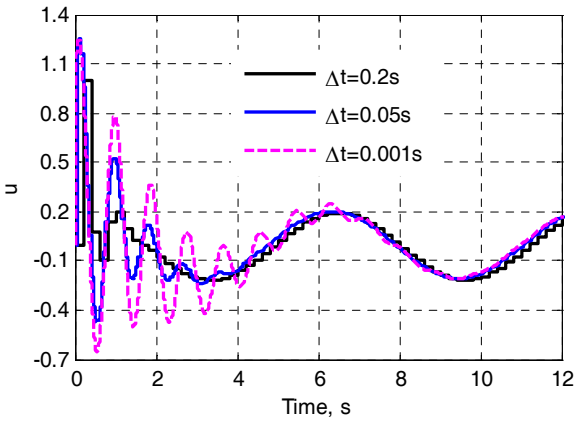

Fig. 4.1 Constraint oscillations from inverse simulation

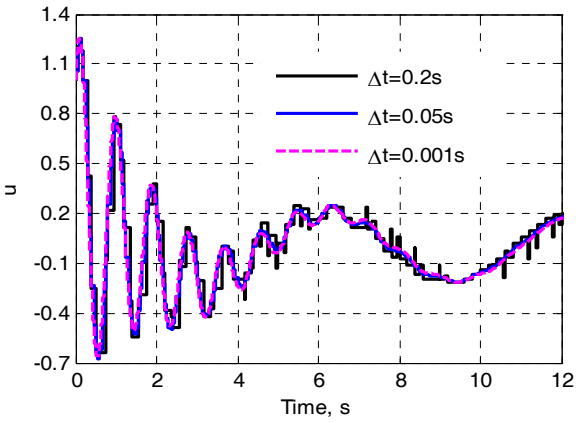

Fig. 4.2 Constraint oscillations from model inversion

Figs. 4.1 and 4.2 both show the constraint-oscillation phenomenon within the early part of the records. In the case of when $\Delta t = 0.001$ s (red line), it can be seen that both methods provide the same results. In addition, compared with model inversion where the variation of the value Δt only slightly affects the accuracy while keeping nearly the same shape, the value of Δt has a bigger influence on the inverse simulation process. With the larger Δt values such as 0.2 s and 0.05 s, the inverse simulation will not provide good results due to the fact that a lot of information that should be present is omitted during the process. This is illustrated in Fig. 4.1 (blue and black lines), in which the result looks very good but are inconsistent with results from the inverse model. All these results show that, in practice, it is necessary to select a smaller Δt in inverse simulation to accurately capture the system dynamic properties as well as to eliminate the high-frequency oscillation (Gao & Hess, 1993). In addition, it also proves that selection of Δt value should be related to the internal dynamic properties of the system as well as to the convergence characteristics of the NR algorithm in the inverse simulation process. Finally, it can be concluded that constraint oscillations may be one of the internal-system properties rather than being associated with numerical issues, as proved in this case. It is not the inverse simulation algorithm which leads to the oscillation or involvement of the pilot (Thomson & Bradley, 1990), but an inherent property of the dynamic system.

This can be further clarified by expanding the output response in the form responding to sinusoidal input:

$$u(t) \approx 0.2\cos(t) + 1.13e^{-0.5t}\sin(7t + \pi/4) \tag{4.4}$$

Equation (4.4) proves that, for the linear case, the constraint oscillation results from the superimposition of sinusoidal functions and functions with exponential coefficients. The second term forms the constraint oscillation component. The oscillation frequency must be decided by the internal-system properties since it is seven times larger than the frequency of the input. Within the first four seconds, the large amplitude and rapidly changing shape of this term significantly influences the inverse simulation process.

In addition, if the value of Δt is greater than 0.4s, for the example system considered, the inverse simulation will not converge, whereas the model inversion technique still can provide good results. The methodologies on which these two methods are based determine this difference. The inverse simulation process involves the NR algorithm, so that larger interval values will lead to a failure to converge. Conversely, the model inversion process implements a forward simulation. There is no convergence problem except that less accurate results are obtained when the larger interval value is adopted. The detailed difference between inverse simulation and model inversion will be presented in a later section of the thesis and is discussed in the papers (Lu *et al*., 2006b; 2007a). If the nonlinear system is considered, the situation will become more complex. However the properties in the vicinity of equilibrium points can be analysed as above.

4.2.2 Relation to the linearised model around trim points

In this subsection, the relationship between the full nonlinear model and linearised representations will be explored and investigated. According to the Hartman-Grobman Theorem (Isidori, 1989), the qualitative properties in the vicinity of isolated equilibrium points are determined by the linearization if the linearization has no eigenvalues on the $j\omega$ axis. This theorem provides a bridge between the nonlinear system model and the equivalent linearised system model. Here, the results of inverse simulation from the nonlinear model and its linearised counterpart around the 80 kts trim point will be compared. However, it should be always kept in mind that during the inverse simulation the process for the nonlinear model diverges from the trim point. Therefore, in the

mathematical sense (as presented by the above Theorem), if the manoeuvres are aggressive or severe, the relationship between these two kinds of models is weak.

In the current application, the linearised model of the nonlinear Lynx helicopter is obtained around the 80 kts. trim point. For the application of inverse simulation to the linear model, selection of an appropriate sampling rate is important. A good choice can guarantee the convergence of the inverse simulation (Lu *et al*., 2006b; 2007a). Following the methodology introduced in Chapter 3, a suitable sampling-rate value can be selected by plotting the magnitudes of zeros versus the variation of sampling rate. The results are shown in Fig. 4.3 and Fig. 4.4.

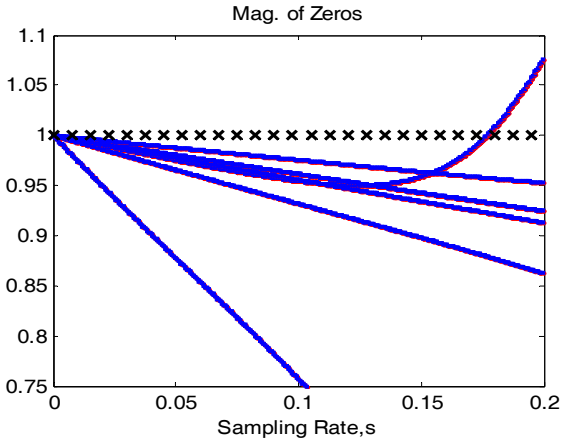

Fig. 4.3 Magnitude of zeros versus sampling rate for Lynx helicopter model (80 kts; zero-order level of the outputs)

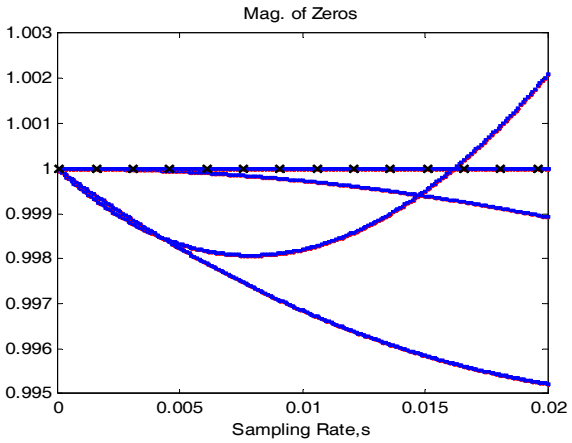

Fig. 4.4 Magnitude of zeros versus variation of sampling rate for Lynx helicopter model (80 kts; second-order level of the outputs)

Fig. 4.3 show the plot of magnitude of zeros versus the sampling rate for the output defined, as normal, in terms of $\begin{bmatrix} x_e & y_e & z_e & \Psi_e \end{bmatrix}$ (i.e. the zero-order level of the outputs). There is an interval from 0 s to 0.18 s in which the magnitudes of all the zeros are less than unity. The selection of sampling rate from this interval can guarantee convergence provided that the NR algorithm shows good numerical characteristics. However, due to the fact that the original inverse simulation package for the nonlinear model is designed based on the output accelerations $\begin{bmatrix} \ddot{x}_e & \ddot{y}_e & \ddot{z}_e & \ddot{\Psi}_e \end{bmatrix}$ (i.e. the second-order level of the outputs), the package for the linearised model is coded in a similar way to provide a fair comparison. The results are shown in Fig. 4.4, where the magnitudes of some zeros are equal to unity over the range of intervals considered. Since inverse simulation is a process that is an approximation to the traditional discretization process as mentioned in Chapter 3, it is still possible to achieve convergence for such a situation if the magnitudes of other zeros are less than unity or around unity. Therefore, a sampling rate of 0.01 s is selected for the linear system with outputs defined in terms of the accelerations. The results from both the nonlinear and linear models are shown in Fig. 4.5 and Fig. 4.6 for the pop-up and hurdle-hop manoeuvres, as introduced in Chapter 2.

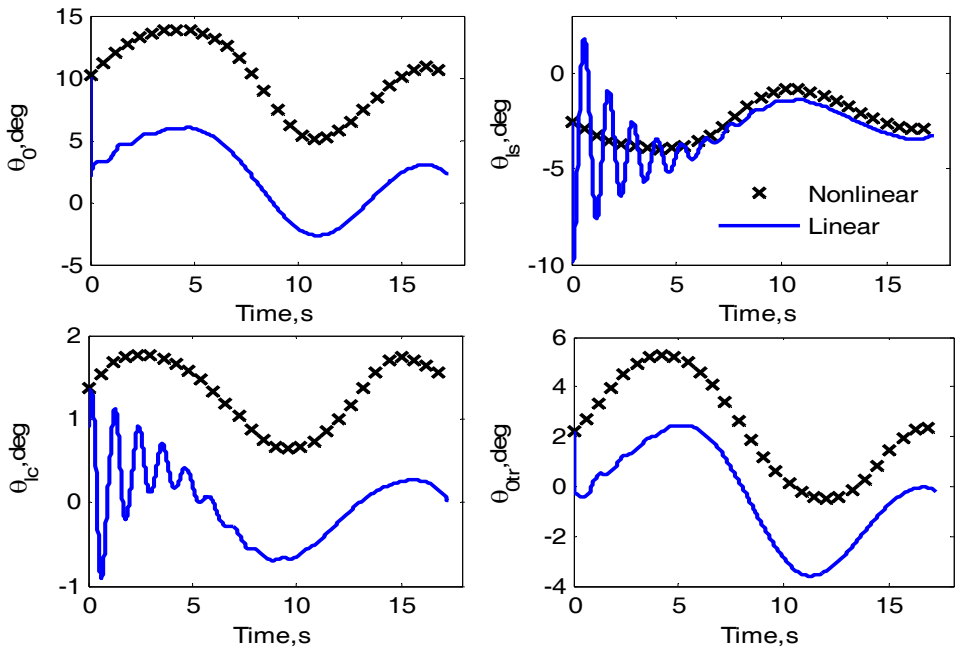

Fig. 4.5 Inputs from hurdle-hop manoeuvre for the Lynx helicopter model (h = 50 m, L = 700 m, and V = 80 kts; Δt = 0.01 s)

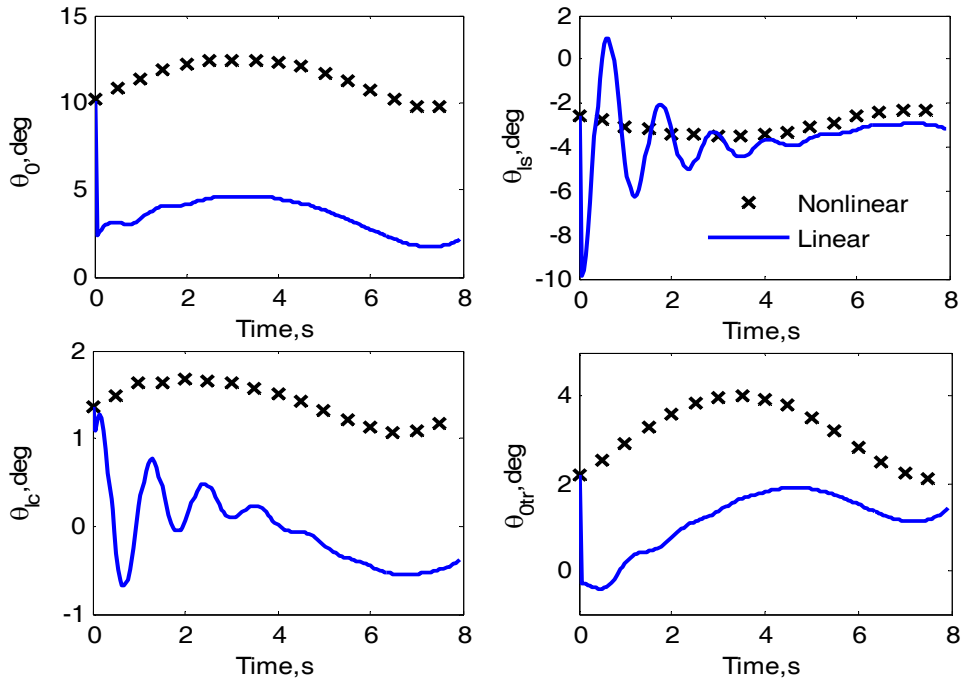

Fig. 4.6 Inputs from pop-up manoeuvre for the Lynx helicopter
(h = 25 m, Lt = 8 s, and V= 80 kts; Δt = 0.01 s)

From Fig. 4.5 and Fig. 4.6 it can be observed that, although the investigated models are different, the shapes of the results are the same except for their amplitudes and the constraint oscillations shown in the case of the linear system model. For the longitudinal cyclic control channel in Fig. 4.5, especially, the second-halves of each of the records are very similar. Fig. 4.5 and Fig. 4.6 both show significant constraint oscillations in the longitudinal and lateral cyclic pitch channels for the linear system. For the other two channels involving the main rotor and tail rotor collective pitch, the two figures show very gentle constraint oscillations. It can be further observed that the frequencies of the constraint oscillations in the linear system are nearly the same (around 5.23 rad/s) regardless of the different manoeuvres. Moreover, this frequency (5.23 rad/s) is close to imaginary part of the complex conjugate zeros $-0.393 \pm 5.43i$ of the linear system which are nearest to the imaginary axis. This is consistent with the conclusion reached in our previous investigation (Section 4.2.1).

The approach described above involves the usual Jacobian linearization of a nonlinear system. More accurate analysis could be possible if the internal system of the nonlinear system is considered, using exact feedback linearization or dynamic inversion (Isidori, 1989). However, most available approaches depend on the existence of the vector relative degree. Therefore, it is impossible to apply them to the nonlinear Lynx helicopter model since the model lacks any analytic relationship between input and output. This is because numerical methods and associated look-up tables have been used to obtain some coefficients within the model.

4.2.3 The influence of sampling rates

It is well know that the sampling rate has a great influence on the discretization process of a continuous system. It may make the discretized system unstable or even show chaotic phenomena. Since the inverse simulation process involves a discretization process, it is believed that the sampling rate has an important role in the inverse simulation and this has already been the subject of much investigation. For inverse simulation based on the integration method with the NR algorithm, a small sampling rate is required to ensure accurate results (Gao & Hess, 1993; Hess *et al*., 1991). However, small values of the sampling rate may well lead to the high-frequency oscillation problem (Lin, 1993). Since the early 1990s, no work has been published that has explored the possibility of the influence of sampling rate on the constraint oscillations. In this subsection, it will be shown how this phenomenon also relates to the sampling rate.

The results from inverse simulation based on the nonlinear Lynx helicopter (Bagiev, 2006) for various sampling rates are presented in Figs. 4.7 to 4.10. Among the results, Fig. 4.7 and Fig. 4.8 show the control efforts from the inverse simulation process and Fig. 4.9 and Fig. 4.10 show the four state variables w (vertical velocity), p (roll velocity), q (pitch velocity), and r (yaw velocity) which display the constraint-oscillations phenomenon.

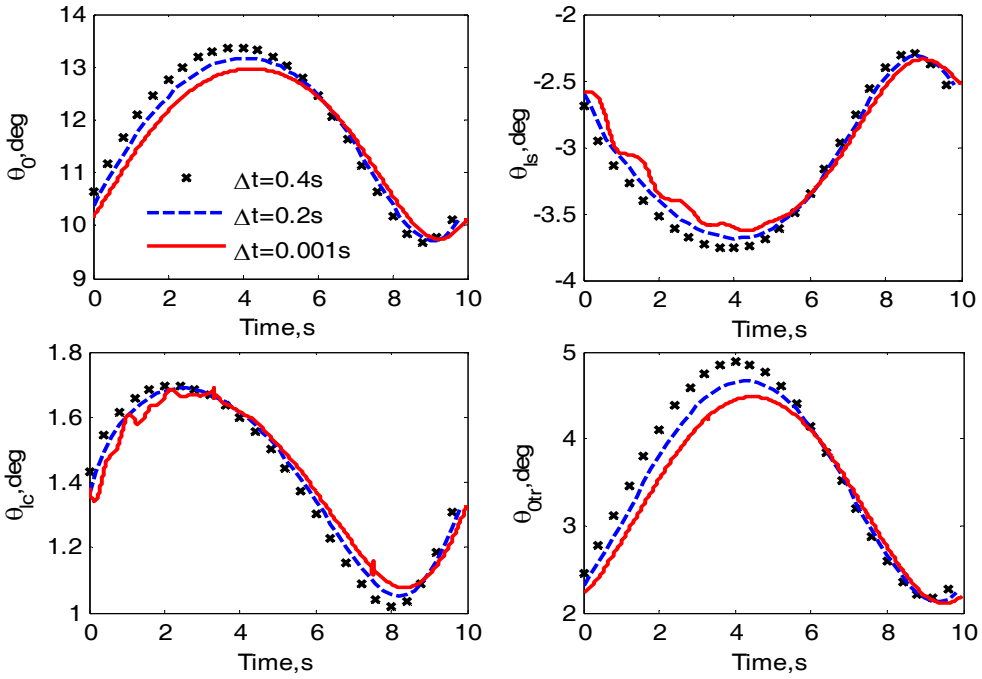

Fig. 4.7 Inputs from pop-up manoeuvre for the nonlinear Lynx helicopter (h = 40 m, Lt = 10 s, and V = 80 kts)

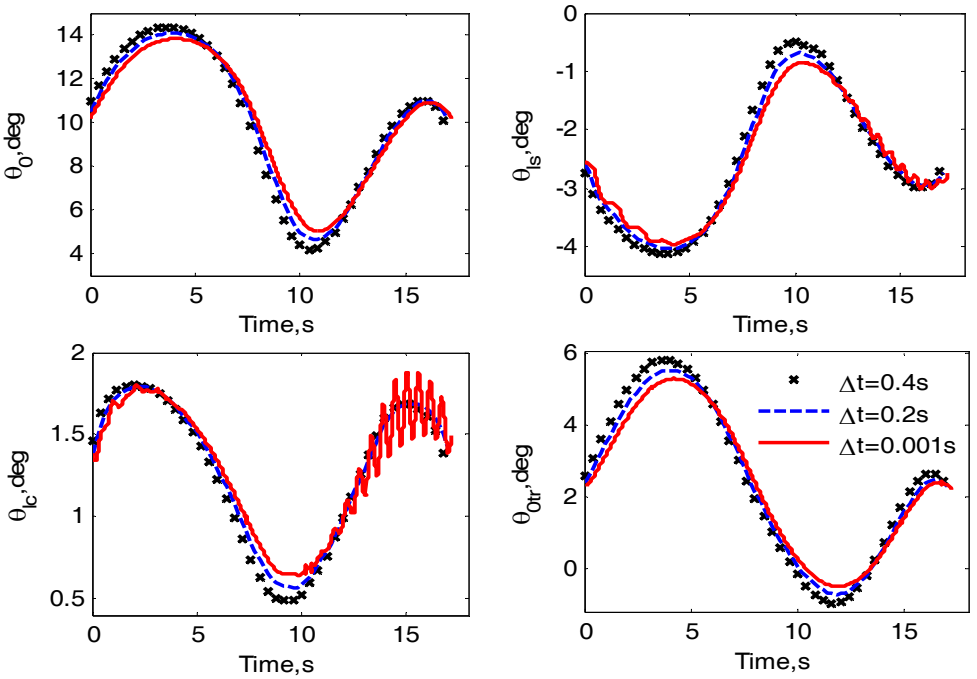

Fig. 4.8 Inputs from hurdle-hop manoeuvre for the nonlinear Lynx helicopter (h = 50 m, L = 700 m, and V = 80 kts)

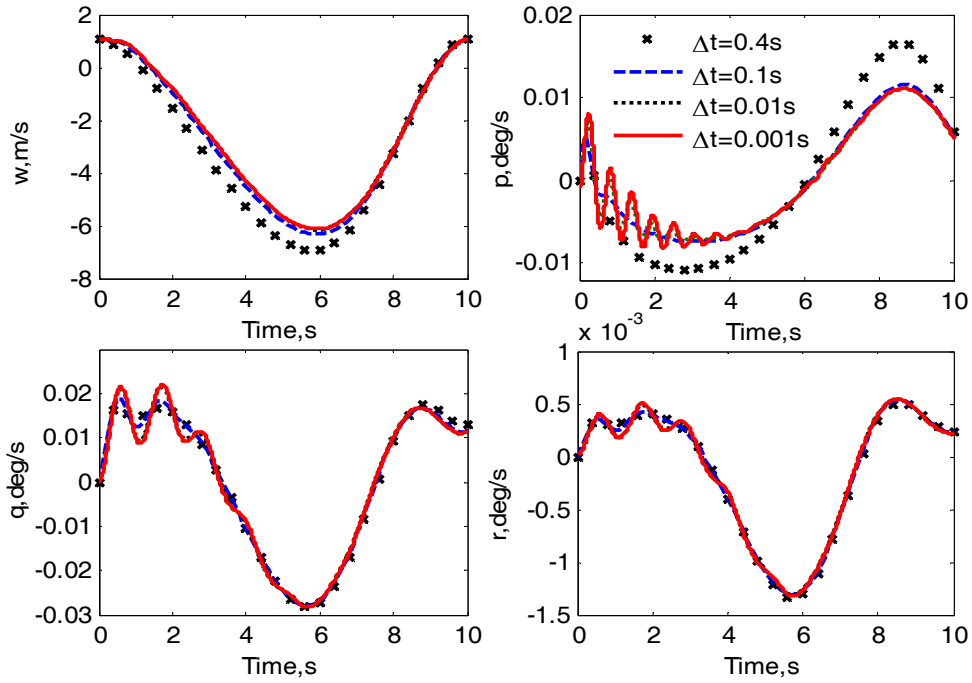

Fig. 4.9 Inputs from pop-up manoeuvre for the nonlinear Lynx helicopter (h = 40 m, Lt = 10 s, and V = 80 kts)

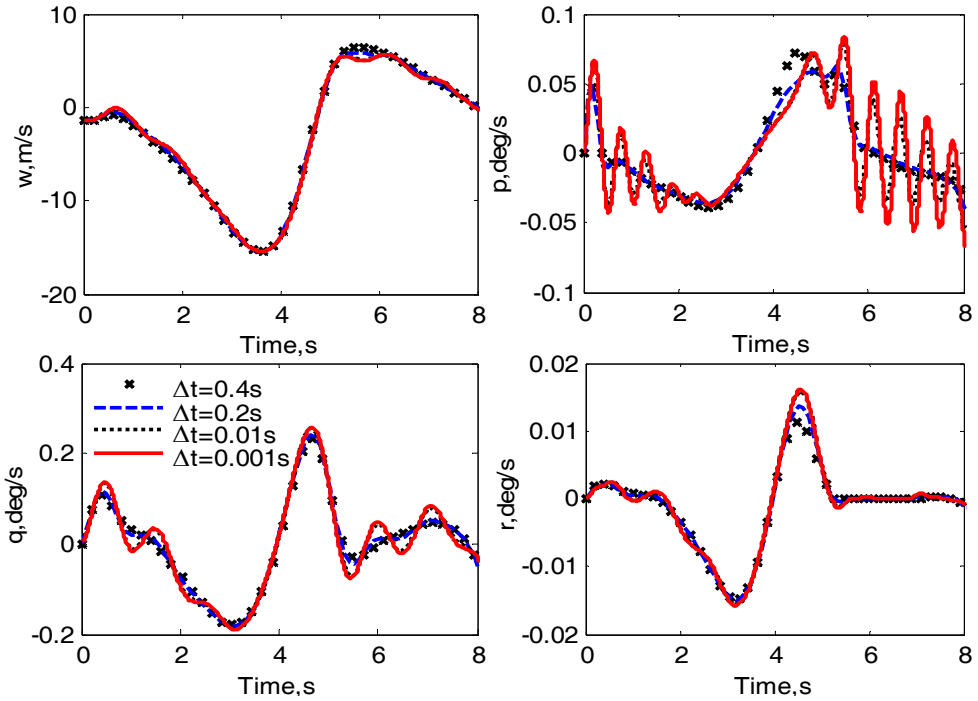

Fig. 4.10 Results from hurdle-hop manoeuvre for the nonlinear Lynx helicopter (h = 20 m, L = 500 m, and V = 120 kts)

where *Lt* in the captions is the total simulation time for the pop-up manoeuvre. The results in Fig. 4.7 and Fig. 4.8 show that the different sampling rates lead to different accuracies. For example, the value of the main rotor collective channel from $\Delta t = 0.4$ s diverges by nearly 15 percent from the one for the case of $\Delta t = 0.001$ s. This demonstrates the need to select a sufficiently small sampling rate to achieve a required level of accuracy. When the value of Δt decreases, the constraint oscillations begin to show in the longitudinal cyclic pitch and lateral cyclic pitch channels. These results are consistent with the linear SISO case shown in Section 4.2.1. Moreover, this phenomenon is shown even more clearly in the pitch and roll channels when $\Delta t = 0.001$ s in Fig. 4.9 and Fig. 4.10.

The constraint-oscillation phenomenon in the linear system has been illustrated in Section 4.2.1. The analysis for the nonlinear system is more complicated. Therefore, here only a general explanation is presented. Since the constraint oscillations are a stable phenomenon which will decay in a small period of time, instability or chaotic phenomena arising from the discretization process could be excluded. The smaller the sampling rate, the closer is the discretized system to the original continuous system. More information over a broader frequency range can thus be included in the system outputs. For the nonlinear system, the information contained in the output consists of information from the model itself and from the inputs as well as from the interaction between input and output. In contrast, the larger the value Δt, the narrower is the frequency bandwidth of interest.

4.2.4 The influence of the manoeuvre

Since the model under consideration is a highly nonlinear system, the inputs have a more significant influence on the output compared with the equivalent linear model. It is therefore worth investigating whether the manoeuvre contributes to the form of the constrained oscillation. Two sets of investigations which have been carried out involve the same manoeuvre with different configurations and different manoeuvres for the same trim condition. The results are shown as follows:

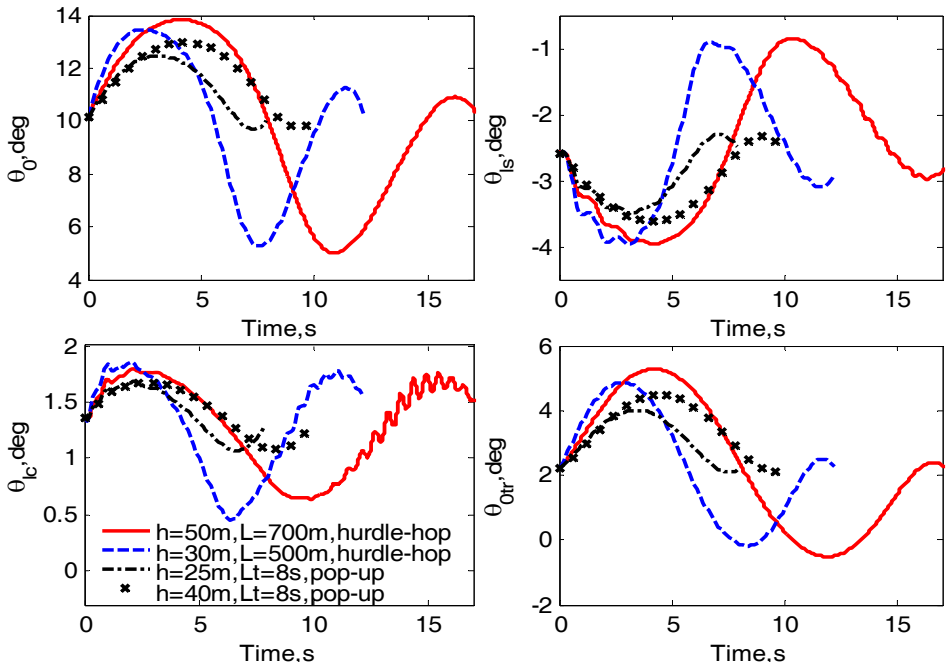

Fig. 4.11 Inputs from hurdle-hop and pop-up for the nonlinear Lynx helicopter (Δt = 0.01 s, V =80 kts)

From Fig. 4.11, it can be observed that the type of input (manoeuvre) does influence the output of the inverse simulation even for the same kind of manoeuvre with slightly different forms. This can be seen from the hurdle-hop manoeuvre with the configuration ($h = 50$ m, $L = 700$ m). The results from this manoeuvre show some constraint oscillations in the early part of the record and very significant constraint oscillations in the second half of the time period in the lateral cyclic channel (θ_{lc}), compared with the other case of ($h = 30$ m, $L = 500$ m). The latter only shows slight oscillations. The results from the pop-up manoeuvres with two different forms are nearly the same and they show constraint oscillations only in the first half of the time period in the longitudinal cyclic (θ_{ls}) and lateral cyclic (θ_{lc}) channels. The same results from these four manoeuvres for the early part of each record possibly relate to the internal dynamics being investigated.

4.2.5 The influence of the trim points

In this subsection, the constraint oscillations at the different trim points are investigated. This investigation aims to check the consistency of the presence of this phenomenon as

operating conditions change. The results from three trim points (40 kts, 80 kts, and 120 kts) are shown as follows:

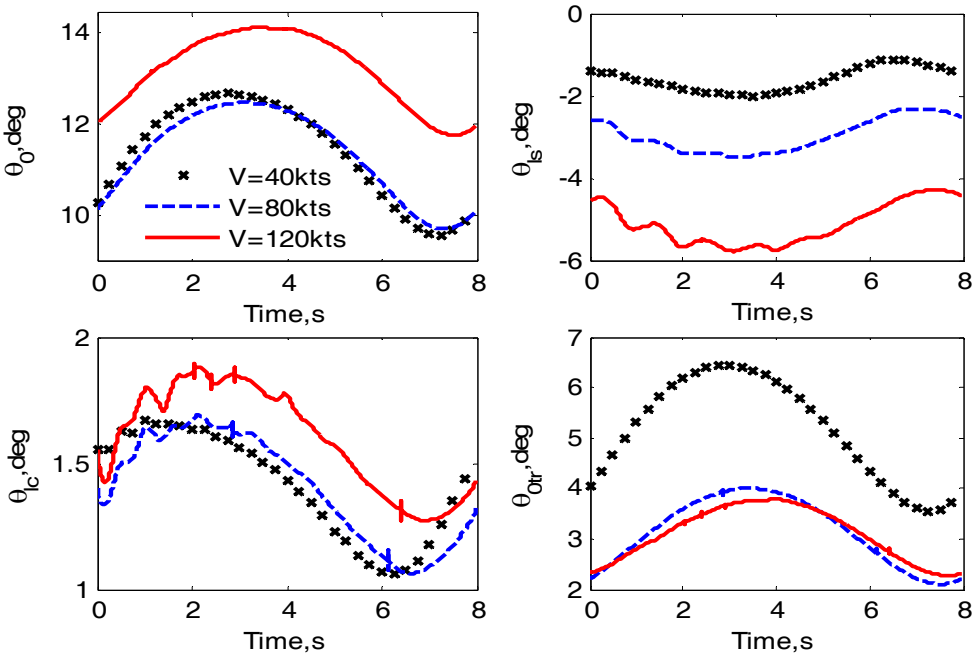

Fig. 4.12 Inputs from pop-up for the nonlinear Lynx helicopter (Δt = 0.0005 s, Lt = 8 s, h = 25 m)

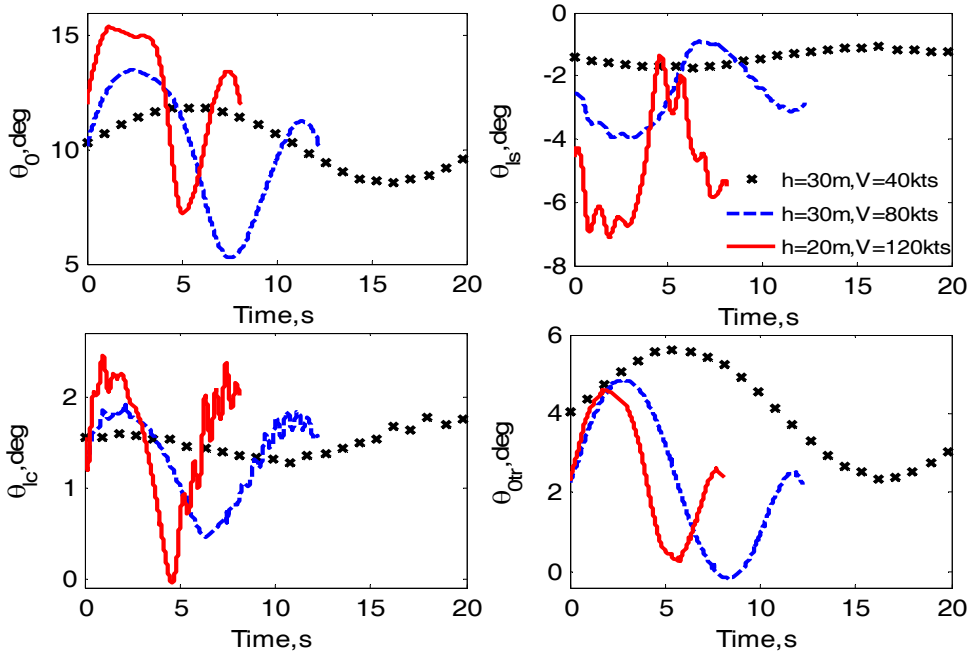

Fig. 4.13 Inputs from hurdle-hop for the nonlinear Lynx helicopter (Δt =0.0005 s, L =500 s)

The constraint oscillations appear in the first half of the time period in the longitudinal cyclic (θ_{ls}) and lateral cyclic (θ_{lc}) channels for each test, as shown in Fig. 4.12 and Fig. 4.13. The internal dynamics of the investigated model may contribute to this. Compared with the results from the other two trim points, the inputs from the trim point at 120 kts show significant constraint oscillations. This probably originates from the fact that this is the highest-speed condition concerned and therefore it involves the shortest time to cover the same span (500 m). In other words, this manoeuvre is the most severe.

4.3 A new method for calculation of the Jacobian matrix

Investigations by Lin (1993) showed the high-frequency oscillation phenomenon with a relatively small value of Δt for the case of a simple linear SISO system. His work has had a considerable influence on the development of inverse simulation and also for its application in that later authors believed that small values of Δt will lead to instability of the inverse simulation process. In this subsection, a new approach has been implemented to calculate the Jacobian matrix and the results from application to Lin's model show that it can effectively eliminate the high-frequency oscillations.

The traditional approach to calculate the Jacobian matrix has been presented in Eq. (2.44). This method only has a local meaning around the $\boldsymbol{u}_j$ point and does not smoothly distribute in the input space. Its accuracy also depends on the size of $\Delta\boldsymbol{u}_j$. This may introduce some inaccuracies. However, it is possible to construct a Jacobian matrix which has global meaning. If the system in Eq. (2.1) has vector relative degree $\boldsymbol{r} = [r_1, \quad r_2, \quad \cdots \quad r_q]^T$ (Isidori, 1989; Sastry, 1999), the above nonlinear system can be written in Eq (4.5) as shown in Appendix-A:

$$\boldsymbol{y}^{(r)}(t) = \begin{bmatrix} \boldsymbol{y}_1^{r_1} \\ \vdots \\ \boldsymbol{y}_q^{r_q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_f^{r_1}\boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{L}_f^{r_q}\boldsymbol{h}_q \end{bmatrix} + \boldsymbol{A}(\boldsymbol{x}) \begin{bmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_q \end{bmatrix} \tag{4.5}$$

where $\boldsymbol{A}(\boldsymbol{x})$ is defined as:

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} L_{\boldsymbol{g}_1}(L_f^{r_1-1}\boldsymbol{h}_1) & \cdots & L_{\boldsymbol{g}_q}(L_f^{r_q-1}\boldsymbol{h}_1) \\ \vdots & \ddots & \vdots \\ L_{\boldsymbol{g}_1}(L_f^{r_q-1}\boldsymbol{h}_q) & \cdots & L_{\boldsymbol{g}_q}(L_f^{r_q-1}\boldsymbol{h}_q) \end{bmatrix} \tag{4.6}$$

If the vector $\boldsymbol{y}^{(r)}$ is considered as the new ideal output vector and $\boldsymbol{u}$ remains the input vector then Eq. (4.5) represents the direct relationship between input and the new output. Thus the Jacobian matrix becomes:

$$\frac{\partial \boldsymbol{y}^{(r)}}{\partial \boldsymbol{u}} = \boldsymbol{A}(\boldsymbol{x}) \tag{4.7}$$

It has been assumed that the original system has vector relative degree $\boldsymbol{r}$ and thus $\boldsymbol{A}(\boldsymbol{x}_0)$ is non-singular (where $\boldsymbol{x}_0$ is the initial condition). This means that $\boldsymbol{A}(\boldsymbol{x})$ is also non-singular in the neighbourhood of $\boldsymbol{x}_0$. In practice, additional constraint conditions will be included within the inverse simulation process. For applications involving small amplitude manoeuvres, the global meaning of $\boldsymbol{A}(\boldsymbol{x})$ in the state variable manifold is usually assured. As a result, the stability and accuracy of the inverse simulation process is improved. The advantages are more apparent if the linear system is considered in that its Jacobian matrix is constant, whereas the implementation of Eq. (2.44) leads to a cumbersome form and reduced accuracy.

Now consider Lin's model shown in Eq. (4.8) as an example.

$$\ddot{x}(t) + x(t) + x(t) = \sin(2t) \quad x(0) = 0 \text{ and } \dot{x}(0) = 0 \tag{4.8}$$

After transforming the above model into the standard state space and letting $x = \boldsymbol{x}_1$ and $u = \sin(2t)$ form the following equation is obtained:

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 &= -\boldsymbol{x}_1 - \boldsymbol{x}_2 + u \end{aligned} \tag{4.9}$$

This system has relative degree two ($r = 2$). Its Jacobian matrix is a constant matrix, shown in Eq. (4.10), according to Eq. (4.7). In addition, the direct relationship between the input and the output has been derived as shown in Eq. (4) in Lin's paper. Therefore, based on this equation the Jacobian matrix takes the form of Eq. (4.11) if the traditional method is

adopted. From this equation, it is evident that the accuracy of the Jacobian matrix depends on the size of the sampling interval Δt.

$$J = A(x) = \frac{\partial \dot{y}}{\partial u} = 1 \tag{4.10}$$

$$J = \frac{1}{\omega_n^2}\{1 - e^{-\sigma \cdot \Delta t}[\cos(\omega_d \cdot \Delta t) - \frac{\sigma}{\omega_d}\sin(\omega_d \cdot \Delta t)]\} \tag{4.11}$$

where $\sigma = 0.5$ and $\omega_d = \sqrt{1 - 0.5^2}$. The inverse simulation results with Eq. (4.10) are shown in Fig. 4.14, where the fourth-order RK integration method is used for the forward simulation, with a step size $\Delta t/60$. The results for Eq. (4.11) are shown in Fig. 4.15, in which the Euler method is adopted for the forward simulation with a step size $\Delta t/60$.

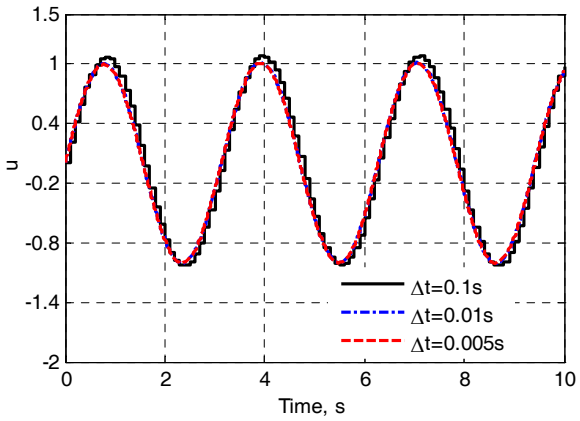

Fig. 4.14 Inverse simulation with the modified Jacobian matrix

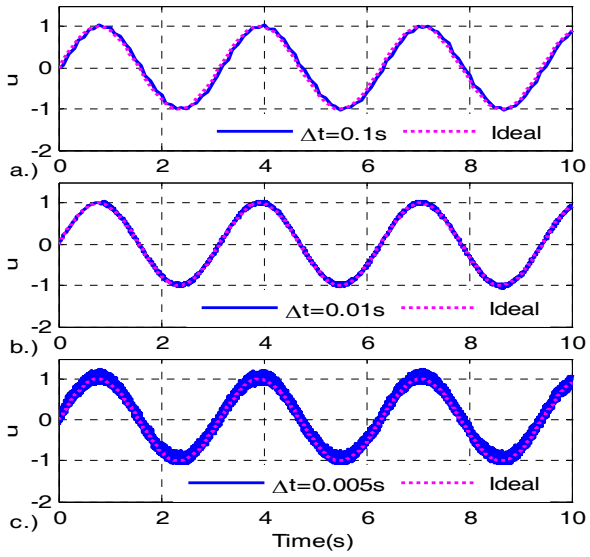

Fig. 4.15 Inverse simulation with the traditional Jacobian matrix calculation

Figure 4.14 shows a perfect set of inverse time histories without high-frequency oscillations even with the small interval value Δt =0.005 s. The results demonstrate that numerical stability has been achieved and are completely different from the ones in Lin's paper, where the high-frequency oscillations appear with the smaller discretization intervals, as illustrated in Fig. 4.15. In addition to this, the simulation process shows that the method presented here is far faster than the traditional approach. As shown in Fig. 4.15, there is evidence of high-frequency oscillations when the value Δt is reduced, but the shape of these oscillations is not exactly the same as those given in Lin's paper. In the current application, Lin's results could not be reproduced precisely. However, the good results shown in Fig. 4.14 have already proved that the numerical stability of the inverse simulation could be improved by finding a better approach to calculate the J matrix.

This method also can be applied in the helicopter field. In fact, the approach that demonstrated improved numerical stability through defining the output in the acceleration level (Rutherford & Thomson, 1996) is a simplified application of the technique presented here, where the authors gave more weight to the physical explanation. The current approach is equivalent to finding a closer analytical relationship between the input and output, or closer to the vector relative degree.

4.4 Summary

The high-frequency oscillation phenomenon and the redundancy problem have been reviewed in this chapter. The constraint oscillations shown in the results from the inverse simulation procedure from a nonlinear system model represent a very complex phenomenon. Various factors, such as the sampling rate, the type of manoeuvres, and the internal dynamics of the model itself, may contribute to the form of this phenomenon. The smaller sampling rate will provide more information in the output and therefore makes the phenomenon more distinct. The severity of the manoeuvre also influences this oscillatory phenomenon, with more severe manoeuvres leading to more distinct oscillations. Moreover, it can be concluded that the internal dynamics plays an important role in constraint

oscillations because of its consistent presence in all the above investigations. Finally, if the input-output analytic relationship can be derived, such as by calculating the vector relative degree, more thorough and accurate knowledge can be obtained and more analysis can be carried out.

Chapter 5

A Sensitivity-Analysis Method for Inverse Simulation

Contents

This chapter discusses some numerical problems mentioned in the previous chapters and a new method based on sensitivity-analysis theory is developed and evaluated to overcome these issues. The work described here has been published in *Journal of Guidance, Control, and Dynamics* (Lu *et al*., 2007c).

5.1 Introduction

Sensitivity analysis is a well-known methodology which has been used extensively in reliability engineering, electrical circuit theory, and control engineering. In an editorial, Saltelli and Scott (1997) present twelve persuasive reasons to explain why and where SA should be considered. In spite of its simplicity, Ratto (2001) shows that SA can help to evaluate the effectiveness of predictions and identify the most important aspects of a model. Sato, Ueda and Ohmori (2004) have applied SA to determine the key parameters from hundreds of parameters for a conventional time varying simulation model. Appendix-C

gives an account of the implementation of this approach to identify the parameters for the inverse Autogyro longitudinal model from simulated data. However, no application of SA to the inverse simulation field appears to have been published to date.

Inverse simulation is a technique that generates the expected forward control inputs such that the mathematical model of a vehicle can follow a prescribed path. The classification of inverse simulation methods and numerical issues existing in the integration-based approach have been discussed in Chapter 4. This chapter describes the first application of SA theory to the problems of inverse simulation for dealing with the above issues and describes a new technique that has been termed 'inverse sensitivity' (Rosenwasser & Yusupov, 2000). Many ideas from sensitivity analysis theory are incorporated into this method and this provides opportunities for a more thorough investigation of system properties through the inverse simulation process.

This new approach developed in this chapter aims to deal more effectively with the following three issues:

I. The approximation method to calculate the Jacobian matrix. Traditionally, if the Jacobian matrix cannot generally be determined analytically, approximation methods are used to determine partial derivatives of output variables with respect to input variables (Hess *et al*., 1991). However, this will inevitably reduce the accuracy.

II. The high-frequency oscillations arising from the integration-based method. This phenomenon may possibly be due to a number of factors, such as step tolerance limits, initial conditions, choice of sampling rate, unobservable states, and multi-solutions etc, as been discussed in detail in Chapter 4.

III. The redundancy problem associated with the number of inputs being greater than the number of outputs, due to the non-square nature of the Jacobian matrix.

The chapter begins by introducing the mathematical development of the new SA method. The theory on which it is based is also presented, including discussion of numerical stability and convergence properties. Comparisons with a traditional integration-based inverse simulation method are presented using results from an application of inverse simulation to a relatively simple fixed-wing aircraft model and a six-degree-of-freedom nonlinear model of a Lynx helicopter.

5.2 Development of the new method

This section involves the derivation of the SA-based approach for inverse simulation. In addition, the proposed method will be compared with the traditional integration-based technique in order to show the different strengths and weaknesses of the two approaches.

5.2.1 Derivation of the algorithm

In the aircraft field of application, inverse simulation is usually carried out from a trimmed flight condition (equilibrium point) with a suitable initial input $\boldsymbol{u}_0$, for the system represented in Eqs. (2.33) and (2.34). In addition, the inverse simulation process is ideal and it is assumed that no other factors such as exogenous disturbances and measurement noise are involved, as been discussed in detail in Chapter 2. Hence, the system output $\boldsymbol{y}$ can be viewed as being the result of the variation of the input vector $\boldsymbol{u}$. Now, if the input vector $\boldsymbol{u}$ is regarded as a set of time-varying parameters $\boldsymbol{\alpha}$, which is certainly independent of the state variables, this is a determining set of parameters because initial conditions (equilibrium points) of the system under investigation are assumed constant in the inverse simulation process. Moreover, $\boldsymbol{\alpha}$ is *complete* since all state variables of the investigated model can be uniquely determined with the given inputs during the ideal simulation process (Rosenwasser & Yusupov, 2000). Therefore, the parameter set $\boldsymbol{\alpha}$ is the unique factor which determines future variations of all state-variable values because of its determining and *complete* properties. As a result, Eqs. (2.33) and (2.34) can be modified to take the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\alpha}) \tag{5.1a}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\alpha}) \tag{5.1b}$$

In the traditional inverse simulation algorithm, the input vector is assumed constant within the time interval $t_k < t \le t_{k+1}$. Therefore, it follows that the vector $\boldsymbol{\alpha}(t)$ will be a constant parameter vector $\boldsymbol{\alpha}_k$ in this time interval. In the following context, this interval is focused upon and $\boldsymbol{\alpha}_k$ is replaced by $\boldsymbol{\theta}$.

For the system given in Eqs. (5.1a) and (5.1b) it can be shown that within the time interval $t_k < t \le t_{k+1}$ the sensitivity function

$$\boldsymbol{Z}(\boldsymbol{\theta}, t) = \frac{\partial \boldsymbol{x}(\boldsymbol{\theta}, t)}{\partial \boldsymbol{\theta}^T} \tag{5.2}$$

exists, provided (Rosenwasser & Yusupov, 2000):

1) the initial conditions for a general case depend uniquely on $\boldsymbol{\theta}$:

$$\boldsymbol{x}_k = \boldsymbol{x}_k(\boldsymbol{\theta}, t_k) \tag{5.3}$$

2) the variables in Eq. (5.3) are continuously differentiable with respect to $\boldsymbol{\theta}$,
3) the solution of $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\theta}, t)$ which satisfies Eq. (5.3) is also continuously differentiable with respect to $\boldsymbol{\theta}$ and
4) the functions $\boldsymbol{f}$ and $\boldsymbol{g}$ are continuously differentiable with respect to their arguments.

From Eq. (5.3) it follows that:

$$\frac{d\boldsymbol{x}_k}{d\boldsymbol{\theta}^T} = \frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{\theta}^T} + \dot{\boldsymbol{x}}_k \frac{dt_k}{d\boldsymbol{\theta}^T} \tag{5.4}$$

Since t_k is a constant, the initial condition for the sensitivity equation shown as Eq. (5.2) is given by:

$$\boldsymbol{Z}(\boldsymbol{\theta}, t_k) = \frac{d\boldsymbol{x}_k}{d\boldsymbol{\theta}^T} \tag{5.5}$$

In addition, it can be shown that if the conditions for Eq. (5.2) hold then, for a sufficiently small perturbation vector $\boldsymbol{\mu}$ of the vector $\boldsymbol{\theta}$, there exists a first-order approximation for $\boldsymbol{x}(t, \boldsymbol{\mu} + \boldsymbol{\theta})$ within $t_k < t \le t_{k+1}$ as follows:

$$\Delta \boldsymbol{x}(t,\boldsymbol{\mu}) = \boldsymbol{x}(t,\boldsymbol{\mu}+\boldsymbol{\theta}) - \boldsymbol{x}(t,\boldsymbol{\theta})$$
$$\Delta \boldsymbol{x}(t,\boldsymbol{\mu}) \approx d\boldsymbol{x}(t,\boldsymbol{\theta}) = \sum_{i=1}^{q} \boldsymbol{Z}_i(t,\boldsymbol{\theta})\boldsymbol{\mu}_i \tag{5.6}$$

where q is the dimension of the $\boldsymbol{\theta}$ vector.

By transforming Eqs. (5.1a) and (5.1b) into the sensitivity-function form, it follows that:

$$\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\alpha}^T} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}^T} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\alpha}^T} \tag{5.7a}$$

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\alpha}^T} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}^T} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}^T} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\alpha}^T} \tag{5.7b}$$

If the following equations are defined:

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{V} \quad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T} = \boldsymbol{A}(t) \quad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{B}(t)$$

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{H} \quad \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}^T} = \boldsymbol{C}(t) \quad \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\alpha}^T} = \boldsymbol{D}(t)$$

then Eqs. (5.7a) and (5.7b) can be expressed in the simplified form:

$$\dot{\boldsymbol{V}} = \boldsymbol{A}(t) \cdot \boldsymbol{V} + \boldsymbol{B}(t)$$
$$\boldsymbol{H} = \boldsymbol{C}(t) \cdot \boldsymbol{V} + \boldsymbol{D}(t) \tag{5.8}$$

Equations (5.8) are the continuous sensitivity equations, which provide an alternative approach to calculate the system output sensitivity function $\boldsymbol{H}$ by a process involving forward simulation.

Since the parameter set $\boldsymbol{\theta}$ is *complete* ($\boldsymbol{\alpha}$ is *complete*) in the interval $t_k < t \le t_{k+1}$ the disturbance $\Delta\boldsymbol{\theta}$ results in the output variation $\Delta\boldsymbol{y}_{k+1}$. Hence, the inverse simulation process becomes an inverse problem for finding the value $\Delta\boldsymbol{\theta}$ from the following equation:

$$\Delta \boldsymbol{y}_{k+1} = \Gamma(\Delta\boldsymbol{\theta}) \tag{5.9}$$

where Γ represents the relationship between the variation $\Delta\boldsymbol{y}_{k+1}$ and the disturbance $\Delta\boldsymbol{\theta}$. In this paper, the method of *solution by inspection* (Rosenwasser & Yusupov, 2000) is adopted because it can apply to the situation where the relationship between $\Delta\boldsymbol{\theta}$ and $\Delta\boldsymbol{y}_{k+1}$ in Eq. (5.9) is *incorrectly* posed. The approximate value for $\Delta\boldsymbol{\theta}$ is found from an

intermediate quantity $\Delta\boldsymbol{\gamma}_k$ in the region around $\boldsymbol{\theta}$ such that the distance between $\Delta\boldsymbol{y}_{k+1}$ and $\boldsymbol{g}(\Delta\boldsymbol{\gamma}_k)$ is minimal. Hence

$$\Delta\boldsymbol{\theta} \approx \Delta\boldsymbol{\gamma}_k = \sum_{n=0}^{l} \Delta\boldsymbol{\gamma}_k^{(n)} \tag{5.10}$$

$$\boldsymbol{\theta} \approx \boldsymbol{\gamma}_k^{(l)} = \boldsymbol{\gamma}_k^{(0)} + \Delta\boldsymbol{\gamma}_k \tag{5.11}$$

where l is the number of iterations, and $\boldsymbol{\gamma}_k^{(0)}$ is the initial condition for $\boldsymbol{\gamma}_k$.

Now a relationship can be defined as follows:

$$\boldsymbol{E} = \boldsymbol{y}_{d,k+1}(\boldsymbol{\theta}) - \boldsymbol{y}(\boldsymbol{\gamma}_k^{(n+1)}) \tag{5.12}$$

Using the results above, the term $\boldsymbol{y}(\boldsymbol{\gamma}_k^{(n+1)})$ in Eq. (5.12) can be replaced with its first-order sensitivity-function approximation. Furthermore, for inverse simulation in the context of aircraft and similar engineering applications (such as robotics), the conditions required are usually satisfied due to the fact that the nonlinear mathematical model of the aircraft (or other engineering system) and the defined output manoeuvre are continuous and smooth:

$$\boldsymbol{E} = \boldsymbol{y}_{d.k+1}(\boldsymbol{\theta}) - [\boldsymbol{y}(\boldsymbol{\gamma}_k^{(n)}) + \boldsymbol{H}_k^{(n)}\Delta\boldsymbol{\gamma}_k^{(n)}] = \Delta\boldsymbol{y}_{k+1}^{(n)} - \boldsymbol{H}_k^{(n)}\Delta\boldsymbol{\gamma}_k^{(n)} \tag{5.13}$$

where $\Delta\boldsymbol{y}_{k+1}^{(n)} = \boldsymbol{y}_{d,k+1}(\boldsymbol{\theta}) - \boldsymbol{y}(\boldsymbol{\gamma}_k^{(n)})$. To update Eq. (5.13), the current values $\boldsymbol{H}_k^{(n)}$ and $\boldsymbol{\gamma}_k^{(n)}$ have to be available in advance. Here it is possible to separate the calculation of $\boldsymbol{H}_k^{(n)}$ and $\boldsymbol{\gamma}_k^{(n)}$ by individually calculating $\boldsymbol{H}_k^{(n)}$ from Eq. (5.8) with the current value $\boldsymbol{\gamma}_k^{(n)}$, especially when the direct analytic input-output relationship of Eq. (5.1b) cannot be found. For such cases this method could show advantages in that it allows $\boldsymbol{H}_k^{(n)}$ values to be found directly by a smooth process involving a traditional one-step ($t_k < t \le t_{k+1}$) forward simulation of Eq. (5.8) with the initial conditions of Eq. (5.5), which determines sufficient conditions for the existence of $\boldsymbol{H}_k^{(n)}$. This approach is quite similar to the traditional integration-based method in that the latter calculates outputs with the current iterative input through one-step forward simulation. In conclusion, the proposed approach could obtain the Jacobian matrix through simulation instead of by means of the approximation (Hess *et al*., 1991).

It should be noted that the implementation of the proposed approach to calculate $\boldsymbol{H}_k^{(n)}$ increases the computational demands since the order of Eq. (5.8) is q times larger than that of the original system represented by Eqs. (5.1a) and (5.1b). Therefore, there is a need to balance the increase in the time required against the improved accuracy. If the analytic input-output relationship exists or can be constructed by finding the vector relative degree in advance (Appendix-A), the traditional approach remains a candidate. The investigations in Chapter 4 have shown that with the constructed input-output relationship, the traditional inverse simulation approach provides greater stability and faster convergence than was previously believed possible. It can also overcome the problem found by Lin (1993). The approach of Rutherford and Thomson (1996), which successfully eliminates the high-frequency oscillations, belongs to this class of methods. In addition, Eq. (5.8) could be suitable for situations when the system [Eqs. (5.1a) and (5.1b)] has no relative degree (Ramakrishna *et al*., 2001) or where high accuracy is required.

Regardless of the method selected, state variables are updated by a one-step forward simulation process. Then, it is reasonable to define an LQ performance index $\boldsymbol{\Pi}$:

$$\boldsymbol{\Pi}_k^{(n)} = \int_{t_k}^{t_{k+1}} (\Delta \boldsymbol{Y}_{k+1}^{(n)} - \boldsymbol{H}_k^{(n)} \Delta \boldsymbol{\gamma}_k^{(n)})^T \boldsymbol{W} (\Delta \boldsymbol{Y}_{k+1}^{(n)} - \boldsymbol{H}_k^{(n)} \Delta \boldsymbol{\gamma}_k^{(n)}) dt \tag{5.14}$$

where $\boldsymbol{W}$ is a weighting matrix. By calculating the derivative of $\boldsymbol{\Pi}$ with respect to $\Delta \boldsymbol{\gamma}_k^{(n)}$, it is possible to find the minimum condition for Eq. (5.14):

$$\frac{\partial \boldsymbol{\Pi}_k^{(n)}}{\partial \Delta \boldsymbol{\gamma}_k^{(n)^T}} = \int_{t_k}^{t_{k+1}} (\boldsymbol{H}_k^{(n)^T} \boldsymbol{W} \boldsymbol{H}_k^{(n)} \Delta \boldsymbol{\gamma}_k^{(n)} - \boldsymbol{H}_k^{(n)^T} \boldsymbol{W} \Delta \boldsymbol{Y}_{k+1}^{(n)}) dt = 0 \tag{5.15}$$

The following equations can be obtained after simplifying the above equation:

$$\begin{aligned} &\boldsymbol{Q}_k^{(n)} \Delta \boldsymbol{\gamma}_k^{(n)} = \boldsymbol{P}_k^{(n)} \\ &\boldsymbol{Q}_k^{(n)} = \int_{t_k}^{t_{k+1}} \boldsymbol{H}_k^{(n)^T} \boldsymbol{W} \boldsymbol{H}_k^{(n)} dt \quad \boldsymbol{P}_k^{(n)} = \int_{t_k}^{t_{k+1}} \boldsymbol{H}_k^{(n)^T} \boldsymbol{W} \Delta \boldsymbol{Y}_{k+1}^{(n)} dt \end{aligned} \tag{5.16}$$

According to Eq. (5.16), the values $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ can be determined by integration over the interval and the value $\Delta \boldsymbol{\gamma}_k^{(n)}$ can then be found directly. Hence, the parameter $\Delta \boldsymbol{\gamma}_k^{(n)}$ can be updated at each time interval according to the following equation

$$\boldsymbol{\gamma}_k^{(n+1)} = \boldsymbol{\gamma}_k^{(n)} + \boldsymbol{\lambda}_k \cdot \Delta \boldsymbol{\gamma}_k^{(n)} \tag{5.17}$$

and thus $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ can also be updated. In Eq. (5.17), λ_k ($0 < \lambda_k < 1$) is a selectively added multiplier, which can increase the convergence speed.

The algorithm will stop if the value $\Delta\boldsymbol{\gamma}_k^{(n)}$ is less than a threshold value defined in advance. The algorithm will then progress to the next time interval. Hence, the series of values of $\boldsymbol{\theta}$ can be found in the whole time space and these values again form the input vector.

It should be noted that in Eq. (5.16), the SA method suggests an alternative way to avoid the redundancy problem of the control inputs outnumbering the outputs (Hess *et al*., 1991) in that $\boldsymbol{Q}_k^{(n)}$ is calculated by the sensitivity-function matrix multiplied by its transpose. Therefore, the matrix $\boldsymbol{Q}_k^{(n)}$ is always square and this method can be applied to all kinds of systems with arbitrary input and output order. In addition, values of the weighting matrix $\boldsymbol{W}$ can be selected to give priority to particular inputs and outputs. However, this may lead to more complexity and there is a need to balance this against the potential benefits in terms of the objectives.

5.2.2 Convergence rate and stability of the algorithm

The above algorithm has applied the modified approach of *solution by inspection* to the inverse simulation field. Thus the basic rules for the convergence and stability of the *solution by inspection* approach can also be applied here. Rosenwasser and Yusupov (2000) show the convergence rate of this approach in the following $(n+1)_{th}$ cost function:

$$\Pi_k^{(n+1)} = \Pi_k^{(n)} - \boldsymbol{P}_k^{(n)^T} \boldsymbol{Q}_k^{(n)^{-1}} \boldsymbol{P}_k^{(n)} \tag{5.18}$$

In Eq. (5.18), nonsingularity and the symmetric positive definite nature of the matrix $\boldsymbol{Q}_k^{(n)}$ at each iteration step is sufficient to ensure convergence for the iterative process. The symmetric positive definite property can be satisfied by selecting the weighting matrix $\boldsymbol{W}$ to be symmetric positive definite in Eq. (5.16).

In addition, an algorithm with better stability properties, which is based on the *Hadamard inequality*, can be implemented. This method can guarantee convergence of the iterative process.

5.2.3 Comparisons of the SA approach with the NR method

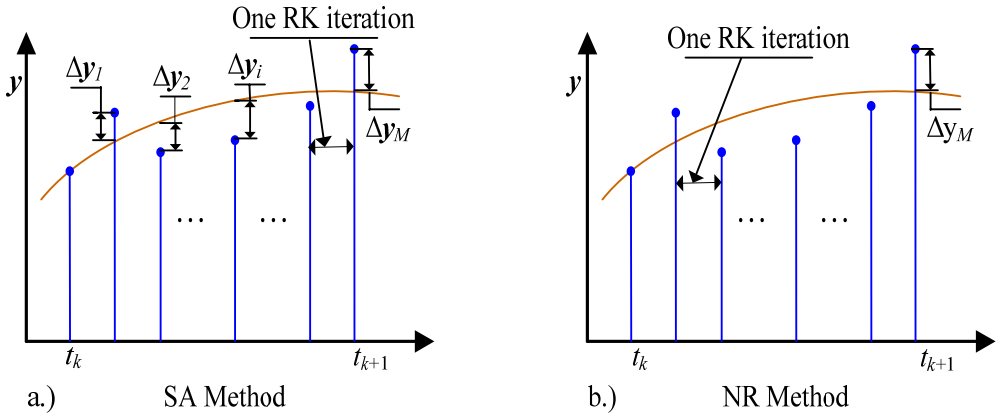

Fig. 5.1 The comparison of information utilization in the k_{th} time interval

Fig. 5.1 shows the main difference between these two methods within the k_{th} time interval. The fourth-order RK algorithm has been selected to carry out the one-step iteration within the interval ($\Delta t/M$). In that period, both methods involve the same number (M) of integrations. However, the integration processes are different in that the SA method will measure the difference $\Delta \boldsymbol{y}_n$ between the ideal value and the value after each one-step RK iteration as shown in Fig. (5.1a). The SA method also calculates the sensitivity-function values after each RK iteration. The expressions for the matrices $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ in Eq. (5.16) include these difference quantities. In contrast, the NR-based inverse simulation process only takes into account the values for the final point after the last one-step RK iteration and ignores the integration process, as shown in Fig. (5.1b). The NR method uses only these final values to update Eq. (2.43). In addition, the structural properties of the matrices $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ provide more information compared with the single Jacobian matrix. The term $\boldsymbol{H}_k^{(n)^T}\boldsymbol{W}\boldsymbol{H}_k^{(n)}$ relates the elements of $\boldsymbol{H}_k^{(n)}$ to each other and the term $\boldsymbol{H}_k^{(n)^T}\boldsymbol{W}\Delta\boldsymbol{Y}_{k+1}^{(n)}$ connects $\boldsymbol{H}_k^{(n)}$ with the deviation from the ideal value.

Although the SA method is similar in some respects to the NR method it should be noted that the SA approach is not equivalent to carrying out an NR-based inverse simulation within a divided time interval ($\Delta t/M$). If this equivalence existed, the NR method would require a different initial value for each divided time interval ($\Delta t/M$). However, here the NR method keeps the assumed input constant over the whole interval Δt in the same way as is done in the SA approach. Moreover, the minimizing optimization of the NR method is based on the difference between the actual and ideal values. However the SA approach does not minimize the output difference and deals instead with the input part $\Delta\boldsymbol{\gamma}_k^{(n)}$ during the inverse simulation process. Finally, the methodologies on which the two approaches are based are completely different. In practice, these results also suggest that further improvements in numerical accuracy may be obtained by increasing the integration number for calculation of $\boldsymbol{Q}_k^{(n)}$ and $\boldsymbol{P}_k^{(n)}$ for situations where the interval Δt is large, thus avoiding the instability discussed by Lin (1993).

5.3 Numerical applications

The case studies selected here relate to two mathematical models. One is a fixed-wing aircraft, the HS125 (Hawker 800) business jet and details of this model can be found in the Appendix-B; the other is a nonlinear Lynx helicopter model (Bagiev, 2006).

5.3.1 Application to a fixed-wing aircraft

The thrust T_t and the elevator angle δ_e act as the inputs in the algorithm implementations for inverse simulation based on the NR and SA methods. The manoeuvres conducted, as been presented in Section 2.4, are the hurdle-hop (Rutherford & Thomson, 1996) and pop-up (Thomson & Bradley, 1990b) manoeuvres in the *z-x* vertical plane and is characterized by the following polynomials, respectively:

$$\begin{aligned} z_d(t) &= 64h\left[(t/t_m)^3 - 3(t/t_m)^2 + 3(t/t_m) - 1\right](t/t_m)^3 \text{ m} \\ V_{f_d}(t) &= 61.87 \text{ m}\cdot\text{s}^{-1} \end{aligned} \tag{5.19}$$

$$z_d(t) = -\left[6(t/t_m)^5 - 15(t/t_m)^4 + 10(t/t_m)^3\right]h \text{ m}$$
$$V_{fd}(t) = 61.87 \text{ m}\cdot\text{s}^{-1} \tag{5.20}$$

where t_m is the time to complete the manoeuvre and h is the height of the obstacle. In this application, the values of h for the hurdle-hop [Eq. (5.19)] and pop-up [Eq. (5.20)] manoeuvres are 70m and 50m, respectively. Furthermore, as shown in Eq. (5.19) and Eq. (5.20), V_f remains constant during the manoeuvre. The quantity t_m for the hurdle-hop manoeuvre is calculated by means of Eq. (2.47). Moreover, in this application, the span distance L for Eq. (2.47) is 1200m. In addition, the value t_m for the pop-up manoeuvre is selected as 10s.

The first priority is to define the calculated manoeuvres in inverse simulation for both methods according to the vector relative degree, if it exists. The implemented model has a vector relative degree [2, 1]. Hence, the implemented manoeuvres in the application of the inverse simulation techniques are defined in terms of acceleration. This guarantees the existence of the input-output analytic relationship. Therefore, the $\boldsymbol{H}$ matrix is available in advance for both methods. For this case, $\boldsymbol{W}$ is selected as a unit matrix and $\boldsymbol{H}$ and $\Delta\boldsymbol{\gamma}$ are calculated as shown in Eq. (5.21). A symbolic differentiation program is used to derive this analytic formulation (Lee & Kim, 1997).

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{1\delta_e} & \boldsymbol{H}_{1T_t} \\ \boldsymbol{H}_{2\delta_e} & \boldsymbol{H}_{2T_t} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \ddot{z}_e}{\partial \delta_e} & \dfrac{\partial \ddot{z}_e}{\partial T_t} \\ \dfrac{\partial \dot{V}_f}{\partial \delta_e} & \dfrac{\partial \dot{V}_f}{\partial T_t} \end{bmatrix} \qquad \Delta\boldsymbol{\gamma} = \begin{bmatrix} \Delta\delta_e \\ \Delta T_t \end{bmatrix} \tag{5.21}$$

In all the following sections the conditions have been defined to give a fair comparison between NR-based and SA-based methods by selecting the same integration number (M) for Eqs. (2.40) and (5.16) within the same interval Δt. All simulations have been run on a *Dell* desktop PC which has a 2.8 G*Hz* processor and a 1 G*byte* memory.

After a first application of these two manoeuvres, the results are shown in Fig. 5.2 and Fig. 5.3:

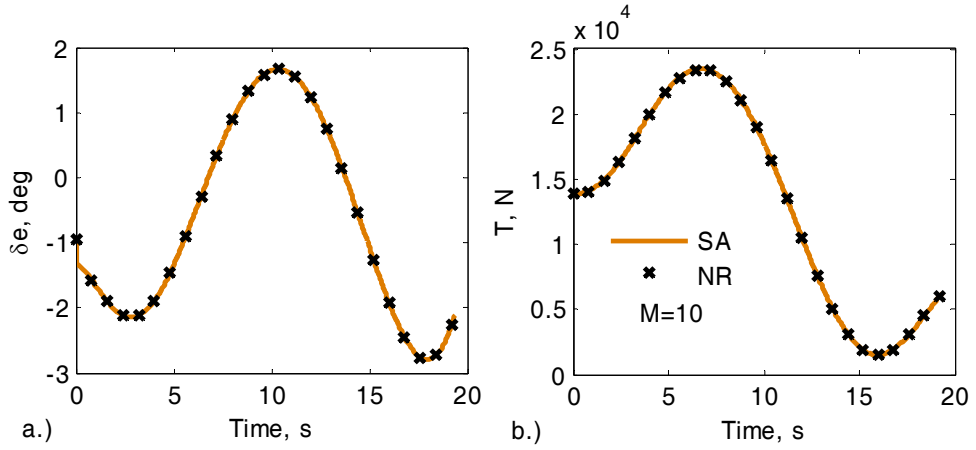

Fig. 5.2 Inverse simulation of hurdle-hop manoeuvre for the HS125 aircraft example (Δt = 0.02 s)

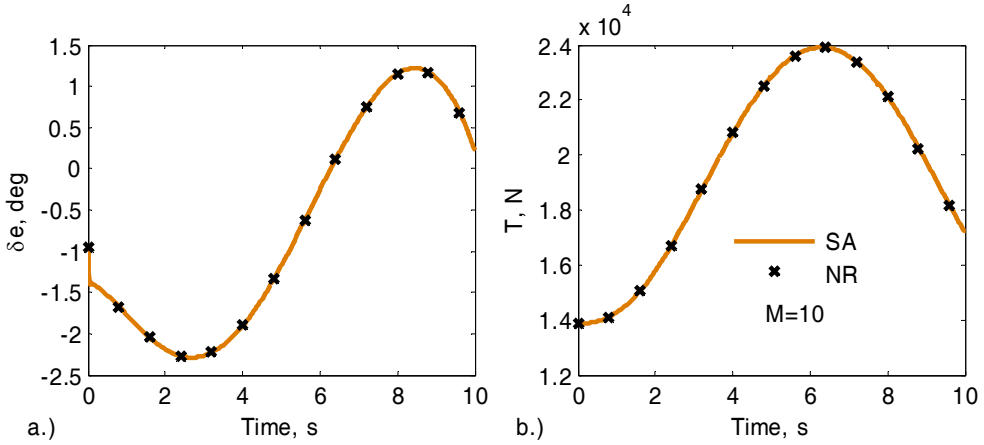

Fig. 5.3 Inverse simulation of pop-up manoeuvre for the HS125 aircraft example (Δt = 0.02 s)

Clearly Fig. 5.2 and Fig. 5.3 prove that the inverse simulation based on the SA method provides the same results compared with the traditional method based on the NR algorithm. The computational time and accuracy for the results are included in Table 5.1 for a variety of different Δt values. Here, the accuracy represented by the variance (*Var.*) describes the consistency of the results from the forward simulation with calculated inputs to the desired manoeuvre. During the simulation, the values for the V_f channel for both approaches are very similar in terms of accuracy. Thus, they are omitted here.

Table 5.1 Computer time and accuracy for the HS125 aircraft example (z_e channel)–M = 10

Manoeuvre		Δt = 0.01, s		Δt = 0.02, s		Δt = 0.025, s		Δt = 0.03, s	
		Time, s	*Var.*	Time, s	*Var.*	Time, s	*Var.*	Time, s	*Var.*
Hurdle-hop	NR	5.6	0.0064	2.6	0.026	2.1	0.0409	1.7	0.0592
	SA	10.5	0.0019	4.4	0.0078	3.4	0.0122	2.8	0.0178
Pop-up	NR	2.7	0.0008	1.3	0.003	1.0	0.0047	0.89	0.0068
	SA	4.5	0.0002	2.1	0.0009	1.6	0.0013	1.3	0.0020

Table 5.1 shows that the results from the SA method can be about four times more accurate than results from the NR method for each time interval Δt. Although this is achieved at the cost of an increase in computation time, it is still acceptable for off-line inverse simulation as the data show.

During the investigation, it was found that both algorithms show a high-frequency oscillation phenomenon within the sampling-rate interval (0.0285 s, 0.0315 s) for $M = 1$. Moreover, values of Δt smaller than the values within this interval can eliminate this problem while larger values can lead to convergence problems for both algorithms for $M = 1$. This phenomenon possibly arises from the fact that the sampling rates lie within the interval for which the algorithms encounter convergence problems. The similar convergence properties for both algorithms for $M = 1$ are probably due to the iterative nature of the calculations in both cases, as shown in Fig 5.1.

It is of interest to investigate whether the high-frequency oscillation phenomenon can be substantially reduced by increasing the value of M. The results from the SA-based method are shown in Fig. 5.4 and Fig. 5.5. In addition, the same series of tests ($M = 1$ to $M = 100$) have been carried for the NR method and they provide results that are, in all cases, exactly the same as the highly oscillatory results shown in Fig. 5.4a and Fig. 5.5a. In other words, increasing the value of M in the NR method has no effect on results. The time required and the accuracy achieved by the two methods is shown in Table 5.2.

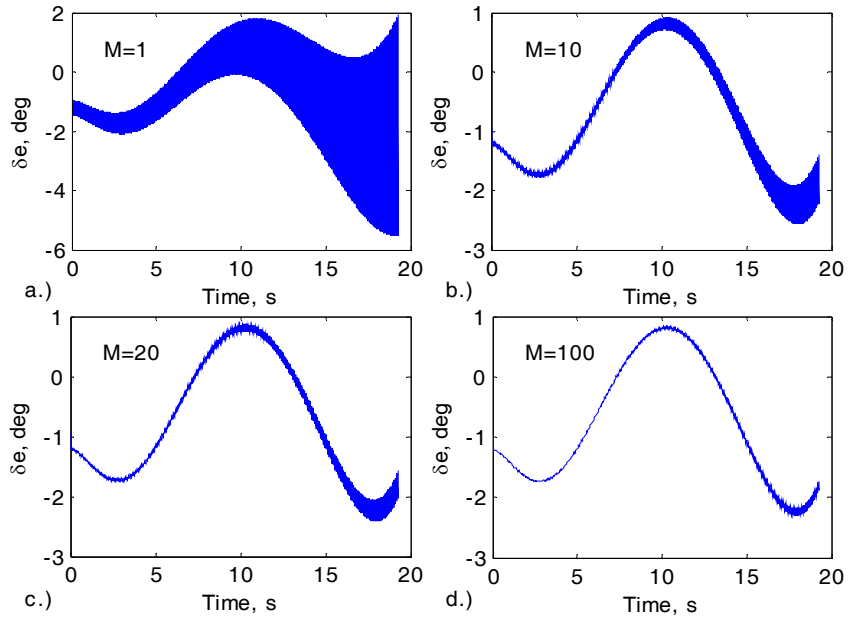

Fig. 5.4 Inverse simulation of hurdle-hop manoeuvre with SA method for the HS125 aircraft example (Δt = 0.031 s)

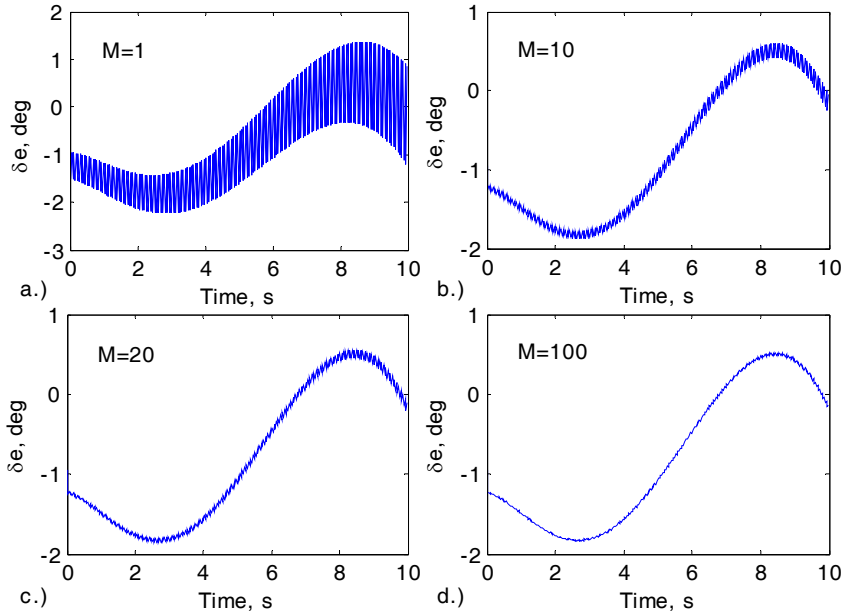

Fig. 5.5 Inverse simulation of pop-up manoeuvre with SA method for the HS125 aircraft example (Δt = 0.031 s)

Table 5.2 Computer time and accuracy for the HS125 aircraft example (z_e channel, Δt = 0.031 s)

Manoeuvre		M = 1		M = 10		M = 20		M = 100	
		Time, s	*Var.*	Time, s	*Var.*	Time, s	*Var.*	Time, s	*Var.*
Hurdle-hop	NR	0.34	0.0632	2.7	0.0632	4.4	0.0632	15.2	0.0632
	SA	0.44	0.0632	1.6	0.0191	7.4	0.0168	26.2	0.0161
Pop-up	NR	0.19	0.0073	0.88	0.0073	1.6	0.0073	7.42	0.0073
	SA	0.22	0.0073	1.28	0.0022	2.4	0.0020	12.5	0.0018

The above results in Fig. 5.4 and Fig. 5.5 show that the high-frequency oscillation phenomenon can be substantially reduced in the SA method with a suitable selection of integration number M for Eq. (5.16). Table 5.2 provides information about the effect of M on the accuracy of results. For the SA method, an accuracy improvement is evident for values from $M = 1$ to $M = 20$. Beyond $M = 20$, these accuracy benefits are still obtained but are less marked and lead to significantly increased computation time. This suggests that there is a trade-off between accuracy and computation time in the choice of M for practical applications. A large value M is preferred but is not essential. In addition, the results also show that it is not essential for the SA method to use a small Δt value to achieve good results, in contrast with the NR approach. Under some conditions, this means that the problems associated with small Δt values which lead to high-frequency oscillations in the NR approach (Lin, 1993) could be avoided. The good performance for the SA approach is probably due to the increase in the information utilized in the inverse simulation process, as discussed in Section 5.2.

5.3.2 Application to a Lynx helicopter model

The model considered in this section is a nonlinear Lynx helicopter developed within the University of Glasgow (Bagiev, 2006). This model involves of five subsystems: fuselage, tail plane, fin, rotor, and tail rotor. The rotor is modelled as an actuator disk using momentum theory and a blade element approach. Similar manoeuvres to the HS125 case (i.e. hurdle-hop and pop-up) are used but with parameters selected that are more appropriate to helicopter flight (Thomson & Bradley, 1998). The four inputs are the traditional four control channels: collective pitch θ_0, longitudinal cyclic pitch θ_{1s}, lateral

cyclic pitch θ_{1c}, and tail rotor collective pitch θ_{0tr}. The four outputs are the three series of equally spaced positional coordinates relative to an earth fixed frame of reference (x_e, y_e, z_e,) and the heading angle Ψ.

The original NR-based software package (Bagiev, 2006) is designed on the basis of the second derivatives of the outputs. This can improve the numerical stability. To provide a useful comparative benchmark, the SA method is also implemented in a similar way. The Jacobian matrix in the original package is calculated by the approximation method. Because some quantities within the Lynx model are obtained by numerical methods and are not expressed in the form of simple analytical relationships in the model equations, it is impossible to calculate the sensitivity functions $\boldsymbol{H}$ through the approach suggested in Eq. (5.7a) and Eq. (5.7b). Therefore, the approximation method is used to calculate the $\boldsymbol{H}$ matrix, which has the following form instead of Eq. (5.8):

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial \ddot{x}_e}{\partial \theta_0} & \frac{\partial \ddot{x}_e}{\partial \theta_{1s}} & \frac{\partial \ddot{x}_e}{\partial \theta_{1c}} & \frac{\partial \ddot{x}_e}{\partial \theta_{0tr}} \\ \frac{\partial \ddot{y}_e}{\partial \theta_0} & \frac{\partial \ddot{y}_e}{\partial \theta_{1s}} & \frac{\partial \ddot{y}_e}{\partial \theta_{1c}} & \frac{\partial \ddot{y}_e}{\partial \theta_{0tr}} \\ \frac{\partial \ddot{z}_e}{\partial \theta_0} & \frac{\partial \ddot{z}_e}{\partial \theta_{1s}} & \frac{\partial \ddot{z}_e}{\partial \theta_{1c}} & \frac{\partial \ddot{z}_e}{\partial \theta_{0tr}} \\ \frac{\partial \ddot{\Psi}}{\partial \theta_0} & \frac{\partial \ddot{\Psi}}{\partial \theta_{1s}} & \frac{\partial \ddot{\Psi}}{\partial \theta_{1c}} & \frac{\partial \ddot{\Psi}}{\partial \theta_{0tr}} \end{bmatrix} \qquad \Delta\boldsymbol{\gamma} = \begin{bmatrix} \Delta\theta_0 \\ \Delta\theta_{1s} \\ \Delta\theta_{1c} \\ \Delta\theta_{0tr} \end{bmatrix} \tag{5.22}$$

After a first application of these two manoeuvres, the results are shown in Fig. 5.6 and Fig. 5.7:

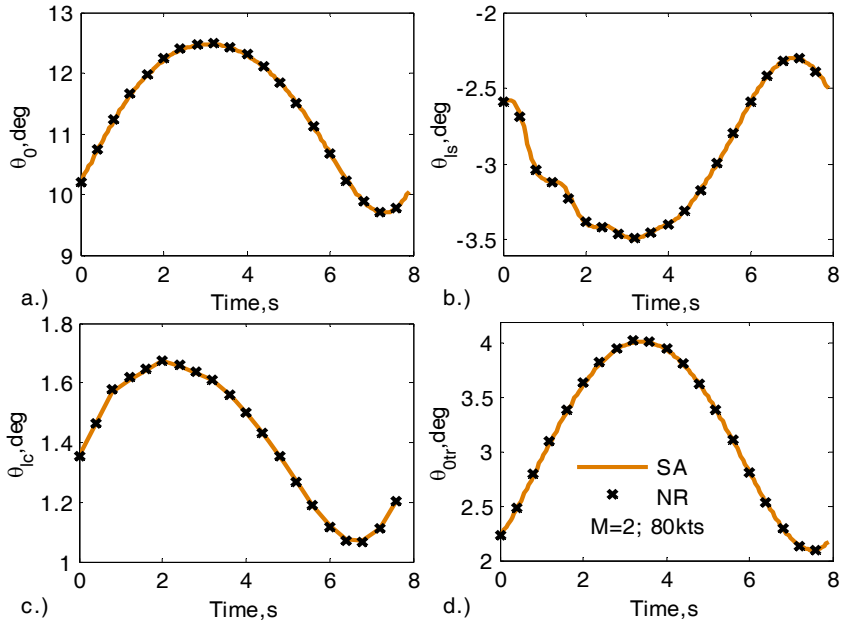

Fig. 5.6 Inverse simulation of pop-up manoeuvre for the Lynx helicopter example (Δt = 0.05 s)

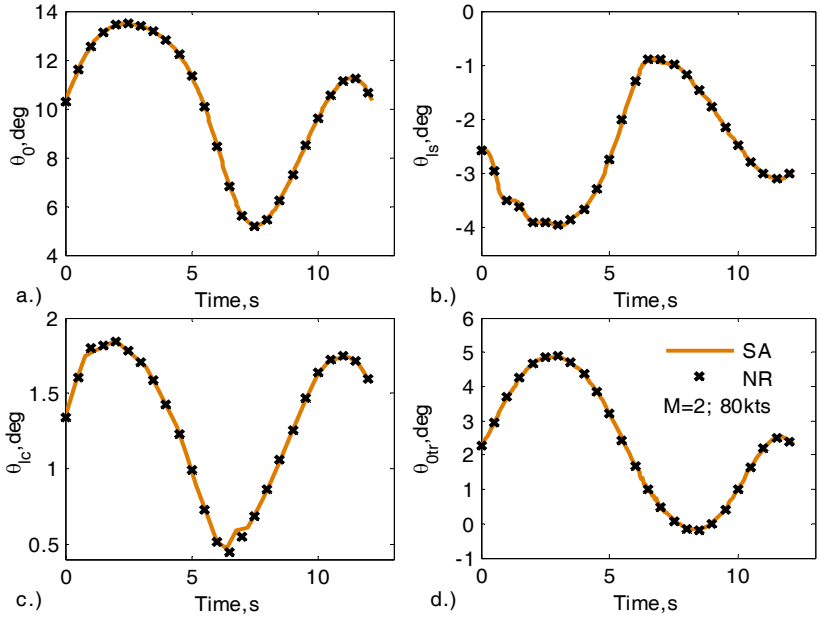

Fig. 5.7 Inverse simulation of hurdle-hop manoeuvre for the Lynx helicopter example (Δt = 0.05 s)

Fig. 5.6 and Fig. 5.7 show that the inverse simulation based on the SA method provides almost the same results as the traditional method based on the NR algorithm for the Lynx

helicopter model. The reason that $M = 2$ is chosen is mainly to achieve a reasonable computation time but relates also to the validity of previous statements concerning the choice of M in Section 5.31. The data for the accuracy of the results are included in Table 5.3 for two different Δt values. In Table 5.3, the results for $M = 1$ from the SA and NR methods are ignored because they are exactly the same as for $M = 2$ from the NR method.

Table 5.3 Output accuracy for the Lynx helicopter example (M = 2, 80 kts)

Manoeuvre	Output	x_e, m		y_e, m		z_e, m		Ψ, deg	
	Δt	0.05, s	0.01, s	0.05, s	0.01, s	0.05, s	0.01, s	0.05, s	0.01, s
Hurdle-hop	NR	0.0024	$3.57e^{-5}$	0.0013	$4.26e^{-5}$	0.0534	0.0021	$9.34e^{-10}$	0.0000
	SA	0.0005	$1.72e^{-5}$	0.0008	0.0003	0.0131	0.0005	$3.35e^{-7}$	0.0000
Pop-up	NR	$1.69e^{-5}$	$6.56e^{-7}$	$9.44e^{-5}$	$4.53e^{-6}$	0.0297	0.0012	$3.13e^{-11}$	0.0000
	SA	$3.34e^{-6}$	$6.00e^{-7}$	$6.18e^{-5}$	$4.35e^{-6}$	0.0074	0.0003	$3.78e^{-12}$	0.0000

The manoeuvres adopted here are carried out in the x-z plane. Hence, in comparing the accuracy of the results particular attention must be given to the output channels x_e and z_e. The results leading to the figures in Table 5.3 show, as in the previous examples, that the accuracy can be increased if the value of M becomes larger (although the results for $M = 1$ are not shown in the table). In addition to the benefits from the different M values, Table 5.3 also shows that the accuracy from the SA method can be more than four times that of corresponding results from the NR method for each time interval Δt. It should be noted that comparison of the computational time is of little relevance in this case, since even for the NR method, the inverse simulation process requires hours of computer time for this complex nonlinear model. The time for the SA method is greater than for the NR method but by less than the factor of two.

5.4 Summary

In this chapter a procedure for inverse simulation based on sensitivity analysis has been developed. Its stability and convergence properties have been discussed. This approach provides a new way to calculate the Jacobian matrix by solving a sensitivity equation. Although it involves increased computational complexity this method avoids the

approximations involved in other published approaches. In addition, the new approach can be applied to arbitrarily redundant situations. The simulations with an HS125 aircraft model and a Lynx helicopter model for the hurdle-hop and pop-up manoeuvres show that the new method provides more accurate results than the traditional approach with an acceptable increase of the computational time. In addition, it can deal with the high-frequency oscillation problem that appears in the inverse simulation process by increasing the integration number.

Chapter 6

A Constrained Derivative-free Inverse Simulation Approach

Contents

The requirement for derivative information in the traditional approaches to inverse simulation may reduce their applicability for situations involving discontinuous manoeuvres or input constraints and discontinuities within the model. This chapter presents a new algorithm, based on the constrained NM method of optimisation, for inverse simulation which is derivative-free to overcome these problems. The results from applications to problems in the marine field show that the new method has better convergence and numerical stability properties compared with the traditional approach for cases that include input saturation in the model or involve a discontinuous manoeuvre. The method has been included in the paper submitted from publication in *Control Engineering Practice* (Lu *et al*., 2007a) and in *Simulation Modelling Practice and Theory* (Lu *et al*., 2007d)

6.1 Introduction

In the control field, it is well known that the performance of a controller may be degraded if the control system designer fails to take account of the input saturation effects. These exist

in real physical systems due to inevitable limitations of mechanical or electrical sub-systems (Lan, Chen & He, 2006; Soroush, Valluri & Mehranbod, 2005). Investigation of the effect of input constraints on control system performance has become an active research topic in recent years (Angeli *et al*., 2005). However, in the inverse simulation field, the previous investigations have given particular consideration to situations involving saturation constraints or discontinuities in the model and manoeuvres.

In fact, the limit effects could present a challenge to traditionally established approaches, involving not only the integration-based approaches (Gao & Hess, 1993; Hess *et al*., 1991; Rutherford & Thomson, 1996) but also differentiation-based methods (Kato & Saguira, 1986; Thomson, 1987; Thomson & Bradley, 1997) and the optimisation approaches (Avanzini *et al*., 1999; 2001; Celi, 2000; de Matteis *et al*., 1995; Lee & Kim, 1997; Lu, *et al*., 2007c). All these techniques involve derivative or gradient information since these approaches depend on continuous and smooth properties of the model and the manoeuvre for inverse simulation.

The integration technique based on the NR method and the differentiation approaches both require calculation of the Jacobian matrix. Although the two-timescale method (Avanzini *et al*., 1999; 2001; de Matteis *et al*., 1995) could deal with the input constraints, it may fail if the manoeuvre or the model being investigated is discontinuous since derivative information is required in the calculation of the Hessian matrix. Besides, the application of this approach involves additional complexity when compared with other methods of inverse simulation. The related contents can refer to Chapter 1 and Chapter 2 in which the historic developments as well as the latest contributions made to the inverse simulation field have been discussed.

To avoid the above problems and achieve increased numerical stability with additional physical insight, a new algorithm for inverse simulation based on the constrained NM method, has been developed and is described in this chapter. It is well known that the NM algorithm can handle discontinuities satisfactorily, particularly if they do not occur near the optimum solution (Lagarias *et al*., 1998; Luersen, Le Richem & Guyon, 2004; Nelder &

Mead, 1965). Furthermore, the derivative-free property can facilitate investigation of some of the numerical issues that exist in the more traditional inverse simulation methods. The new proposed method is an approach that combines optimization with the integration method and does not make use of any numerical or analytic gradient information. In addition, to provide a meaningful and illustrative benchmark for comparison and to show the advantages achieved by the newly developed approach, this chapter focuses on comparisons with the integration technique based on the NR method (Hess *et al*., 1991) because of its widespread use.

The chapter will begin by discussing problems met in cases involving input saturation and discontinuous manoeuvres in the integration-based inverse simulation procedure. Then the mathematical development of the new constrained NM method and the theory on which it is based are presented. Finally, comparisons with traditional inverse simulation methods are presented using results from an application of inverse simulation techniques to five ship models for four different types of manoeuvre.

6.2 Problems with input saturation and discontinuous manoeuvres

If saturation of the control input is considered, the convergence characteristics of the inverse simulation algorithm based on the NR approach may not be as simple as the situation without saturation limits. The inclusion of saturation limits allows the inverse simulation to be more closely related to the characteristics of the actuator and other related mechanical or electrical subsystems, thus providing more physical insight but introduces additional difficulties. The problem may be explained by considering Fig. 6.1 which is a block diagram illustrating what happens in the k_{th} discretized interval for the m_{th} iteration.

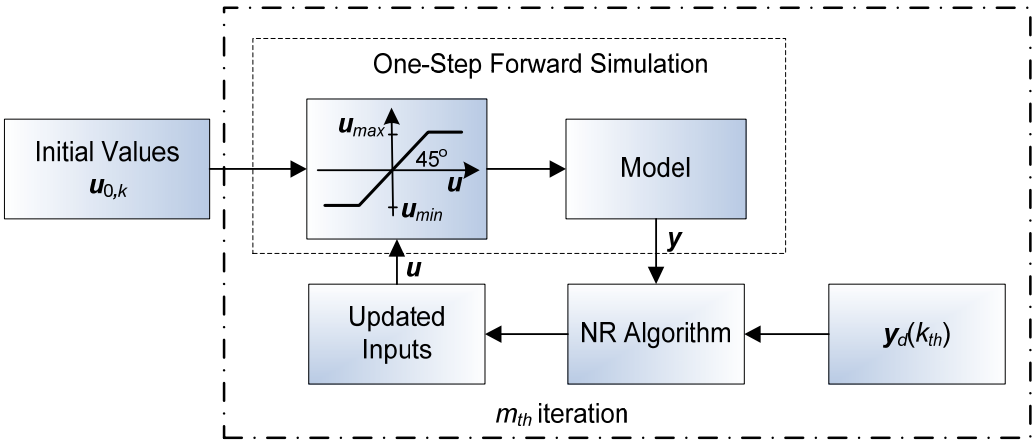

Fig. 6.1 The k_{th} discretized interval of inverse simulation with input saturation

In this diagram the input $\boldsymbol{u}_{0,k}$ represents the initial value of u at the first iteration for the k_{th} discretized interval. When the chosen manoeuvres are demanding, particular subsystems such as actuators and control surfaces may reach their maximum limits. This means that the amplitudes of the inputs that would be required to perform the manoeuvre would be larger than the saturation level $\boldsymbol{u}_{max}$ or $|\boldsymbol{u}_{min}|$. Therefore, if the mathematical model represents the real physical system accurately enough, the amplitudes of the calculated input values would be larger than the saturation levels. However, as shown in Fig. 6.1, before being fed into the model block, the absolute values of input variables have to be limited to $\boldsymbol{u}_{max}$ or $|\boldsymbol{u}_{min}|$. This will make the elements of the corresponding column of the Jacobian matrix zero, as can be seen from Eq. (2.44). Thus, the NR algorithm then fails to converge because the Jacobian matrix is singular. This non-convergence, or no-solution problem, is also consistent with failure of the system to perform such a demanding manoeuvre.

As a consequence, it is more physically meaningful to consider the effects of inclusion of input saturation. However, even when the saturation limits are not reached, the inverse simulation process may not be as simple as the situation without saturation limits. The NR method is known to depend on a proper choice of a starting point for good convergence Cheney & Kincaid, 2004). From the mathematical viewpoint, if $\boldsymbol{u}_{0,1} > \boldsymbol{u}_{max}$ or $\boldsymbol{u}_{0,1} < \boldsymbol{u}_{min}$, the actual values $\boldsymbol{u}$ are equal to $\boldsymbol{u}_{max}$ or $\boldsymbol{u}_{min}$. Therefore, initial values, which the designer considers as good values to assure satisfactory convergence of the inverse simulation

process, may be inappropriate and undesirable in this case with saturation limits present. After a one-step forward simulation, the saturation effect is likely to cause the NR algorithm to fail to converge. Hence, initial values must be selected more carefully in such situations, compared with the case without saturation. Trial and error methods may be necessary in dealing with practical applications.

In addition to failure to converge at the first-step, unexpectedly large inputs occurring during the iterative process may also cause non-convergence. Although the chosen manoeuvre may be assumed to be smooth and not severe, the NR algorithm may give a result $\boldsymbol{u}$ which is larger than $\boldsymbol{u}_{max}$ or $|\boldsymbol{u}_{min}|$ during the iterative process. This computed value may not be abnormal but arises simply as a result of the numerical process. Such a value would not affect the iterative consistency if the model being considered did not include input saturation. However, the inverse simulation process may break down when saturation is included since the updated input has to pass through this nonlinearity. As a consequence, the output from the one-step forward simulation will diverge from what the NR algorithm is 'expecting'.

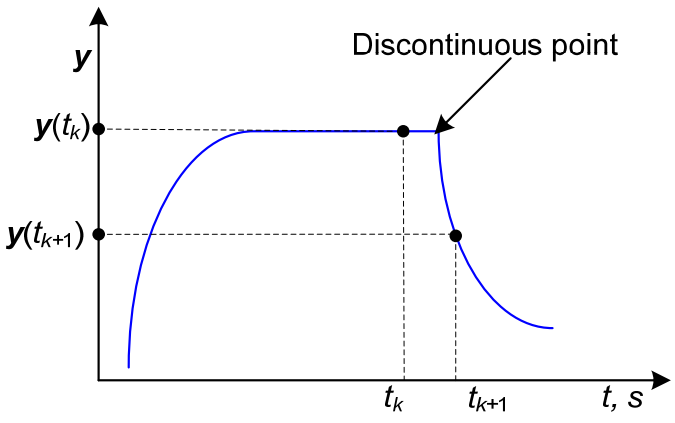

Fig. 6.2 Illustration of a discontinuous point

Finally, a discontinuous manoeuvre can also lead to non-convergence of the inverse simulation. This is because a discretization process is involved in the traditional inverse simulation approach, as illustrated in Fig. 6.2, where a discontinuous point is located within the time interval t_k to t_k+1. When the inverse simulation process meets this special point,

the initial input guess values $\boldsymbol{u}_{0,k}$ and the calculated Jacobian matrix $\boldsymbol{J}$ may not guarantee that a solution can be found for the large transient value $\Delta \boldsymbol{y} = \boldsymbol{y}(t_{k+1}) - \boldsymbol{y}(t_k)$. Therefore, the inverse simulation process cannot converge. In addition, the method proposed by Rutherford and Thomson (1996) may fail since it depends on smoothness properties of the manoeuvre.

6.3 Development of the constrained NM method

All the methods discussed in Chapter 2 introduce additional derivative calculations, such as those associated with the Jacobian matrix or Hessian matrix. However, the direct gradient information is not always available from the model. This issue, as well as other problems existing in the NR method, have been discussed in Chapter 4. For example, values of some parameters in the ship models will depend on environmental factors. Direct search methods, being derivative free and thus avoiding issues associated with discontinuity and input saturation, could provide alternative approaches that might show advantages.

Lewis, Torczon and Trosset (2000) have reviewed the history and development of direct search methods and point out that they remain popular because of their simplicity, flexibility, and reliability. Among direct search methods, the most widely used is the downhill simplex method of Nelder and Mead (1965). It is a popular method for minimizing a scalar-valued nonlinear function of q real variables using only function values, without any derivative information (explicit or implicit). The latest developments of this method (Chelouah & Siarry, 2003; Luersen *et al*., 2004; Wolff, 2004) have expanded its functions so that it can be used to tackle multimodal, discontinuous, and constrained optimization problems but these developments inevitably make the algorithm more complex. The algorithm developed in this chapter is based on the version (Lagarias *et al*., 1998) with an additional input-constrained function (Errico, 2005).

As with the NR method, the NM approach is developed in the interval $[t_k, t_{k+1}]$. One of the distinct differences between the NR and NM methods is that the former updates the input

values by means of Eq. (2.43), but the latter relies exclusively on values of the cost function to find the optimal solution (Lewis *et al*., 2000). Hence, it is important for the NM method to define a good form of the cost function, which may be described by equations of the form:

$$\begin{cases} \min\limits_{\boldsymbol{u}\in\mathbb{R}^q} L[\boldsymbol{u}(t_k)], \quad \text{where} \\ L[\boldsymbol{u}(t_k)] = \sum\limits_{i=1}^{p} \{\boldsymbol{g}_i[\boldsymbol{u}(t_k), \boldsymbol{x}(t_{k+1})] - \boldsymbol{y}_{d_i}(t_{k+1})\}^2 \end{cases} \tag{6.1}$$

subject to

$$\begin{cases} \boldsymbol{u}_{min,j} \le \boldsymbol{u}_j(t_k) \le \boldsymbol{u}_{max,j} \qquad j = 1,2,\ldots,q \\ \dot{\boldsymbol{x}} = \boldsymbol{f}[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)] \end{cases} \tag{6.2}$$

where $L[\cdot]$ is the cost function. If the NM algorithm fails for the quadratic cost-function form of Eq. (6.1), the following equation based on the absolute value can provide an alternative:

$$\begin{cases} \min\limits_{\boldsymbol{u}\in\mathbb{R}^q} L[\boldsymbol{u}(t_k)], \quad \text{where} \\ L[\boldsymbol{u}(t_k)] = \sum\limits_{i=1}^{p} \left| \boldsymbol{g}_i[\boldsymbol{u}(t_k), \boldsymbol{x}(t_{k+1})] - \boldsymbol{y}_{d_i}(t_{k+1}) \right| \end{cases} \tag{6.3}$$

It is not easy to handle the second constrained condition in Eq. (6.2) by the augmented *Lagrangian* method since it includes the first-derivative term. However, this problem can be handled by using the structure of the integration-based approach so that the process to find solutions is divided into two sub-processes: one-forward simulation to obtain $\boldsymbol{x}(t_{k+1})$ and then calculation of the solution $\boldsymbol{u}(t_k)$ from Eq. (6.1) or Eq. (6.3) with the available values $\boldsymbol{x}(t_{k+1})$. The first constrained condition of Eq. (6.2), or even more complicated inequalities, can be handled by the adaptive linear penalty function (Luersen *et al*., 2004). However, this method is quite complicated and unnecessary in the case of the inverse simulation application in that only the input saturation conditions are of interest. For the proposed method in this chapter, the inequalities in Eq. (6.2) are solved by four steps, as shown in the following (Errico, 2005):

Step 1–transformation of input constraints

The purpose of this step is to transform the original domain of the input variables into a new space before searching for the solution using the NM algorithm. The unconstrained input variables will be left alone. If an input variable is constrained by only a lower or an upper bound, a quadratic transformation will be performed by means of the following equations:

$$\begin{cases} \text{if } \boldsymbol{u}_j(t_k) \le \boldsymbol{u}_{min,j} \text{ or } \boldsymbol{u}_j(t_k) \ge \boldsymbol{u}_{max,j} \\ \boldsymbol{u}_{a,j} = 0, \text{ otherwise} \\ \boldsymbol{u}_{a,j} = \sqrt{\boldsymbol{u}_j(t_k) - \boldsymbol{u}_{min,j}} \text{ or } \boldsymbol{u}_{a,j} = \sqrt{\boldsymbol{u}_{max,j} - \boldsymbol{u}_j(t_k)} \end{cases} \tag{6.4}$$

where $\boldsymbol{u}_a$ is the transformed input vector. If both the lower and upper bounds are required, a *sin* transformation can be defined as follows:

$$\begin{cases} \text{if } \boldsymbol{u}_j(t_k) \le \boldsymbol{u}_{min,j} \text{ or } \boldsymbol{u}_j(t_k) \ge \boldsymbol{u}_{max,j} \\ \boldsymbol{u}_{a,j} = -\pi/2 \text{ or } \boldsymbol{u}_{a,j} = \pi/2, \text{ respectively, otherwise} \\ \boldsymbol{u}_{a,j} = 2 \cdot \pi + \arcsin[-1, (1, 2 \cdot \dfrac{\boldsymbol{u}_j(t_k) - \boldsymbol{u}_{min,j}}{\boldsymbol{u}_{max,j} - \boldsymbol{u}_{min,j}} - 1)_{min}]_{max} \end{cases} \tag{6.5}$$

where the added term 2π in Eq. (6.5) is introduced to avoid problems at zero in the NM algorithm. If this is not done the initial simplex is vanishingly small.

***Step* 2–transformation back to the original domain with the constraints**

This step is used to transform the new input domain back into the original domain with the constraints for the each evaluation of the cost function. Thus, the searching domain for input values is based on the values transformed from Eq. (6.4) and Eq. (6.5) and the actual function values evaluated by the NM algorithm have to be obtained by application of a second transformation.

In the approach adopted the unconstrained input variables remain unchanged. For the input variables that are constrained in terms only of a lower or an upper bound, the transformation is applied as follows (Errico, 2005):

$$\begin{cases} \text{For the lower bound}: \boldsymbol{u}_{b,j} = \boldsymbol{u}_{min,j} + \boldsymbol{u}_{a,j}^2(t_k) \\ \text{For the upper bound}: \boldsymbol{u}_{b,j} = \boldsymbol{u}_{max,j} - \boldsymbol{u}_{a,j}^2(t_k) \end{cases} \tag{6.6}$$

where $\boldsymbol{u}_a$ is the transformed or finally calculated input values. If inputs are constrained in terms of both lower and upper bounds, a *sin* transformation can be applied. This transformation is defined as follows:

$$\begin{cases} \text{For the lower and upper bounds}: \\ \boldsymbol{u}_{b,j} = \frac{1}{2} \cdot \{\sin[\boldsymbol{u}_{a,j}(t_k)] + 1\} \cdot (\boldsymbol{u}_{max,j} - \boldsymbol{u}_{min,j}) + \boldsymbol{u}_{min,j} \end{cases} \tag{6.7}$$

The constrained conditions in the cost functions of Eq. (6.1) or Eq. (6.3) may be handled by the above two steps so that the transformed values $\boldsymbol{u}_{b,j}$ are bounded for the NM algorithm.

Step 3–finding of a solution by means of the NM algorithm

A particular form of NM algorithm, which is the modified version described by Lagarias *et al* (1998), is used to find the solutions in the constrained domain but with the function values evaluated in the original domain. It can be summarised as follows. The algorithm first characterises a simplex in q-dimensional space by $q+1$ distinct vertices. Then, based on four rules that involve processes of reflection (ρ), expansion (χ), contraction (γ) and shrinkage (σ), a new point in or near the current simplex is generated at each step of the search. Then a new simplex can be constructed by replacing a vertex in the old simplex, after the function value from Eq. (6.1) or Eq. (6.3) at the new point is compared with the function's values at the vertices of the old simplex. This process is repeated until the diameter of the simplex is less than the specified tolerance. Optimum solutions are thus found for the step under consideration. If each step converges successfully, the complete input time histories $\boldsymbol{u}(t)$ can be formed by combining together the solutions obtained over each interval.

The values of the four important coefficients: ρ, x, γ, and σ used are those recommended by Lagarias *et al* (1998). These are also almost universal choices for the standard NM algorithm and are

$$\rho = 1 \quad \chi = 2 \quad \gamma = 0.5 \quad \sigma = 0.5 \tag{6.8}$$

Step 4– transformation back to the original domain with the constraints

The final solutions from the NM algorithm have to be transformed back to the original domain. This process is similar to Step 2.

The complete computational process can be illustrated by the following flow chart:

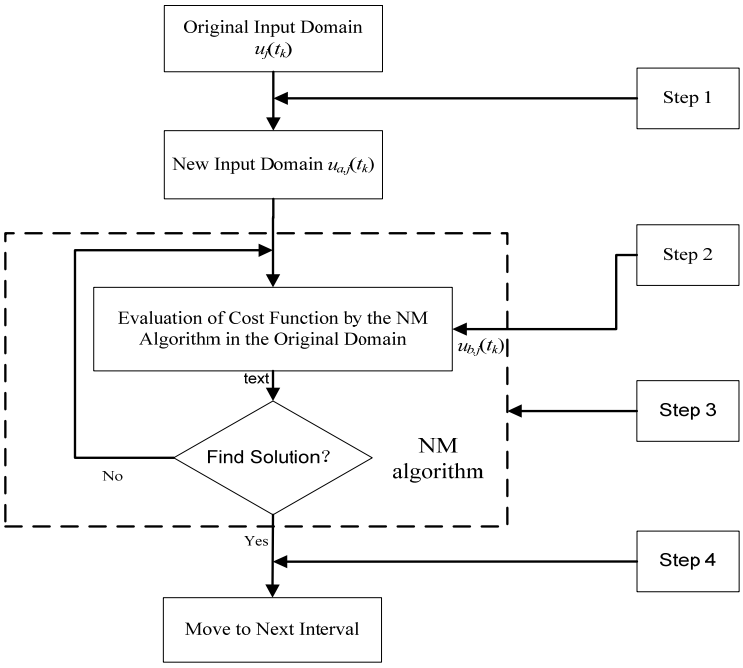

Fig. 6.3 Flow chart for the k_{th} interval of inverse simulation with the constrained NM algorithm

Luersen *et al* (2004) describes a more complicated NM method that can globalise a local search by probabilistic restarts. The essence of this approach is to summarize the topology of the basins of attraction in which a fixed total cost can be reached by a *Gaussian Parzen-Windows* algorithm (Duda, Hart & Stork, 2000). For the method proposed in this chapter, the initial guess values for $\boldsymbol{u}_{k+1,0}$ are the calculated values $\boldsymbol{u}_k$ from the previous step. Thus, if the manoeuvre is smooth and continuous, this could be a good starting point. If it includes discontinuous points, the probability is still high for the NM algorithm to find a global solution because of the narrow searching vector space from $\boldsymbol{u}_k$ to $\boldsymbol{u}_{k+1}$ instead of the whole space $\boldsymbol{u}(t)$. Hence, the problem of the discontinuous point in the manoeuvre discussed in Section 2 may be avoided.

6.4 Numerical examples

To compare the new algorithm with the NR method, five case studies have been selected. They relate to five nonlinear mathematical models – the nonlinear Norrbin model described in Appendix-D, the "Mariner" ship model, a model of a Container ship, a model of a Tanker ship, and a complicated model of an autonomous underwater vehicle (AUV) (Fossen, 1994). These models being considered are different from those considered in most previous inverse simulation investigations in that they include input saturation for each input control channel. Furthermore, for the Norribin, "Mariner", Container, and Tanker ship models, the rudder rate is also constrained, as shown in Fig. 6.4. These limitations degrade the performance of the rudder and result in limited controllability of the system (McGookin *et al*., 2000). The reason for this is due to the fact that the integral term normally included in the controller tends to infinity when the rudder saturates (Donha *et al*., 1998). In addition, some coefficient values are not fixed and change according to the sea conditions. This is another source of discontinuity that increases the problems of inverse simulation.

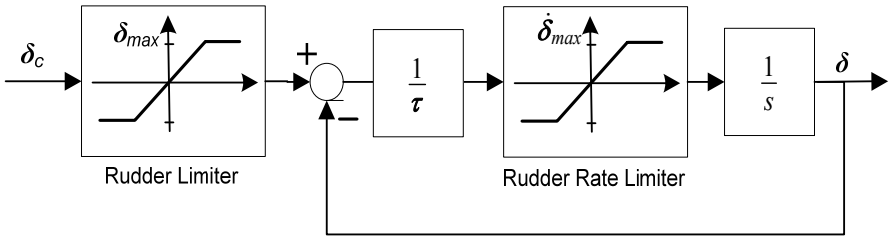

Fig. 6.4 Diagram illustrating rudder amplitude and rate limit (Fossen, 1994)

where the value for the time constant τ is selected to be 1 s. The three kinds of manoeuvres investigated are the turning circle, a zigzag, and a pullout (Lóez *et al*., 2004). To ensure physical meaning and realism, the manoeuvres implemented in this chapter are produced from the ship models themselves. This can avoid issues of non-physical problems discussed in Section 6.2. The pullout manoeuvre, whose shape is similar to the one shown in Fig. 6.2, in fact represents one kind of discontinuous manoeuvre. In addition, due to the symmetry of the pullout manoeuvres in two port and starboard directions, only one of them is carried out as the ideal manoeuvre for inverse simulation.

Table 6.1 Input saturation values for the different ship models

	Norrbin	"Mariner"	Container	Tanker	AUV
δ_{max} , deg	35	40	10	10	20 (five rudder controls)
$\dot{\delta}_{max}$, deg/s	7	5	5	2.33	--
n_{max} , rpm	--	--	160	80	1500

The Norrbin model is a SISO model. The other models are MIMO cases. For example, there are two-output channels in the cases of the "Mariner" and Container models and four outputs for the AUV model. These are determined from physical reasoning taking account of the convergence qualities of the NM and NR algorithms, to describe the manoeuvres for each model. The numbers of input channels for the "Mariner", Container ship, Tanker ship, and AUV models are 1, 2, 3, and 6 respectively, as shown in Table 6.1 (including the rudder-rate). This table also gives the numerical values of the saturation limits for each input channel.

6.4.1 Application to a nonlinear Norrbin model

The details about the Norrbin model are included in Appendix-D. The third equation in Eq. (C.4) there is related to the steering machine structure, as described in Fig. 6.4, where the rudder and rudder-rate limiters are involved in the model. The limiting values shown in Fig. 6.4 are given in Table 6.1.

A third-order reference model, as shown in Eq. (6.9), is used to generate the desired heading response.

$$\frac{\Psi_d}{\Psi_r} = \frac{c_m}{s^3 + a_m s^2 + b_m s + c_m} \tag{6.9}$$

where a_m, b_m, and c_m are constants. In the current application, these values are selected as follows (Unar, 1999):

$$a_m = 0.9341,\ b_m = 0.2040 \text{ and } c_m = 0.0182 \tag{6.10}$$

This choice of reference model can guarantee sufficient smoothness of the heading acceleration.

Inverse simulation has been carried out for a Norrbin type model of the *ROV Zeefakkel* (RZ) ship (Unar, 1999) with the forward speed $U = 10$ m/s and the set heading angles are 20 deg and 50 deg. The quality of the results from inverse simulation has been validated by the feedforward simulation (FFS) with the calculated inputs. This procedure can be described by the following diagram:

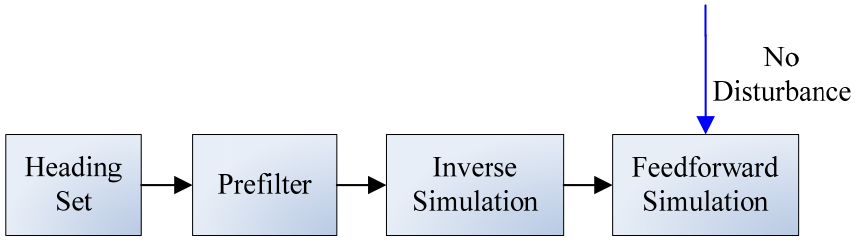

Fig. 6.5 Validation of inverse simulation

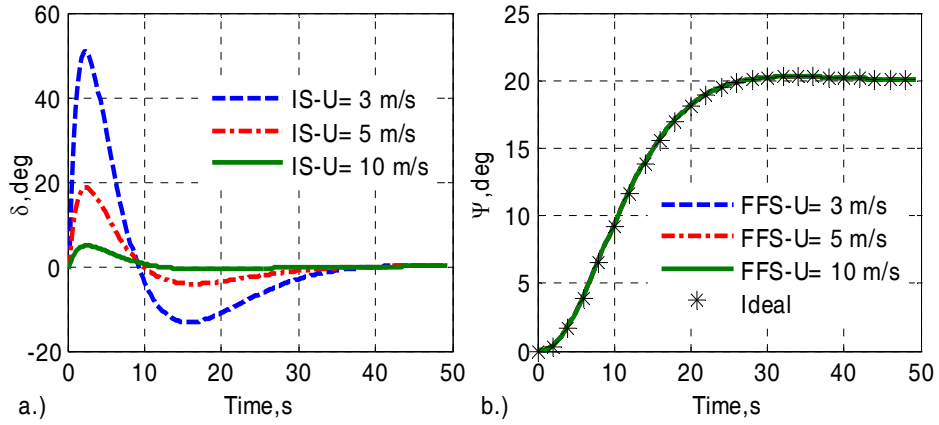

Fig. 6.6 Inverse simulation of the RZ ship without saturation limits (Δt =0.2 s, NR)

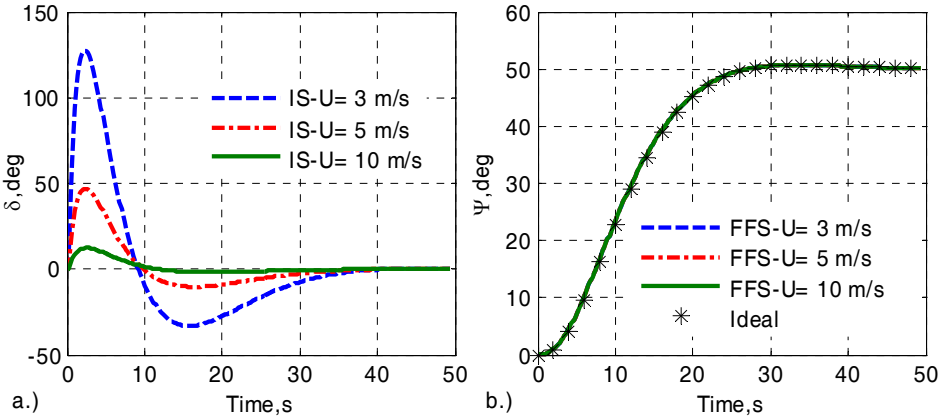

Fig. 6.7 Inverse simulation of the RZ ship without saturation limits (Δt =0.2 s, NR)

As shown in Fig. 6.6 and Fig. 6.7, the results from the inverse simulation and the FFS agree well with each other for both heading angles of 20 and 50 degrees. The right-hand parts of Fig. 6.6 and Fig. 6.7 also contain results from three forward speeds (U = 3, 5, 10 m/s). These results demonstrate the successful application of inverse simulation to the nonlinear RZ ship model without saturation limits. From these figures, it may also be seen that the rudder angles are beyond their limits for part of the time history for the case where $U = 3$ m/s and also for $U = 5$ m/s for a heading angle of 50 degrees. In addition, a further series of tests of inverse simulation based on both the NR and NM algorithms have been successfully run on the RZ model with the forward speeds varying from 1 m/s to 20 m/s. The results are not presented here since they are similar to those shown in Fig. 6.6 and Fig. 6.7.

Inverse simulation also has been investigated on the models with the saturation limits included. As shown in Table 6.1, the rudder limit is 35 degrees and the rudder-rate limit is 7 deg/s. This situation is quite different from the one without the limiters. The inverse simulation based on the NR algorithm fails to converge for the set heading angle of 20 deg if the forward speed U is less than 9 m/s, or for the set heading angle of 50 deg if U is less than 15 m/s. This problem of convergence failure is a feature of the NR algorithm when saturation limits are reached, as discussed in Section 6.2. In this application it arises to a considerable extent from the fact that the inputs or the rate of change of inputs required to track the ideal manoeuvres become larger as the forward speed decreases, as is clearly shown in Fig. 6.6 and Fig. 6.7. In addition, the larger the required heading angle the larger will be the required control effort. Therefore, the case of the set heading angle of 50 deg will show problems of convergence failure at a larger value of speed compared with the case of the set heading angle 20 deg.

However, the inverse simulation based the NM algorithm, which is developed in this chapter to overcome the problem of input saturation, can be used to deal with this kind of situation. Two typical cases with forward speeds 3 m/s and 8 m/s have been investigated. As mentioned above, the NR algorithm failed for both cases. The results from simulations are shown in Fig. 6.8 and Fig. 6.9.

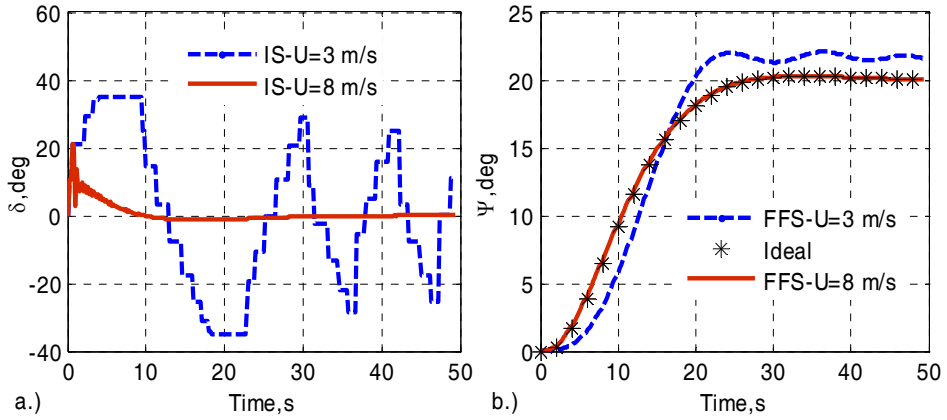

Fig. 6.8 Inverse simulation of the RZ ship with saturation limits (Δt =0.2 s) using the NM algorithm

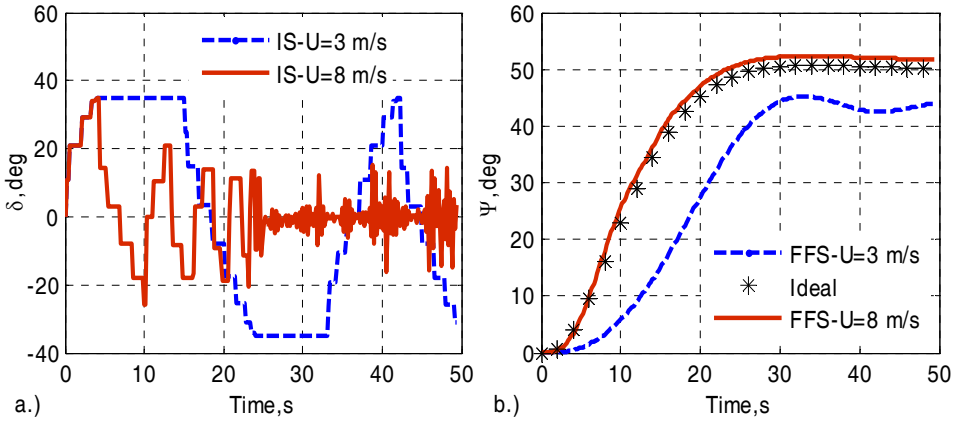

Fig. 6.9 Inverse simulation of the RZ ship with saturation limits (Δt =0.2 s) using the NM algorithm

Fig. 6.8 shows that the NM-based approach achieves good convergence. For a heading angle of 20 deg and the forward speed 8 m/s, the required input is within the saturation limits, as shown in Fig. 6.8a and the trajectory from the FFS also complies well with the ideal manoeuvre in Fig. 6.8b. However, the rate limit value for the rudder on this vessel is 7 m/s and the rates encountered without the limiting reached 34 m/s. Therefore, it is not the rudder amplitude limiter but the rudder-rate limiter that leads the NR-based approach to fail to converge. For a heading angle 20 deg and the forward speed 3 m/s, the amplitude of the required input reaches the saturation level at two periods around the time points 8 and 20 seconds. These saturations further lead to the slight discrepancy between the results of the FFS and the ideal manoeuvre, as shown in Fig. 6.8b. In addition, the input result shows the slowly oscillating shape, which is also shown in the results of the FFS.

The situation becomes worse for the case where the heading angle of 50 deg is considered. Fig. 6.9a shows that the results are affected by high-frequency oscillations after the 22 second time point, although the case for $U = 8$ m/s does converge. However, the smooth results from the FFS comply with the ideal manoeuvre, as shown in Fig. 6.9b. This may be due to the filtering effect of the model, as mentioned in Section 6.2. The case of $U = 3$ m/s is quite challenging since the required inputs are significantly larger. Fig. 6.7 shows that the maximum require input can reach a value of 130 degrees with the minimum of -40 degrees. As a result, the inverse simulation process is in the saturation state during two long periods, shown in Fig. 6.9a and this naturally leads to a large discrepancy from the ideal manoeuvre shown in Fig. 6.9b.

6.4.2 Application to a nonlinear model of the "Mariner" vessel

The parameters configured to generate the manoeuvre are as follows: the time point at which the manoeuvre is started (the time at which the rudder is moved) is 10 s for both the turning circle and the zigzag; the set value for the rudder angle (δ) is 20 deg ($<\delta_{max}$); the cost function is defined, based on Eq. (6.3), as:

$$L[\boldsymbol{u}(t_k)] = \left|\boldsymbol{g}_1[\boldsymbol{u}(t_k), \boldsymbol{x}(t_{k+1})] - \boldsymbol{y}_{d_1}(t_{k+1})\right| + \left|\boldsymbol{g}_2[\boldsymbol{u}(t_k), \boldsymbol{x}(t_{k+1})] - \boldsymbol{y}_{d_2}(t_{k+1})\right| \tag{6.11}$$

The two-output channels for Eq. (6.11) for the turning circle are the surge velocity (u) and the sway velocity (v), which are variables defined in the body-axis system. The reason for using the body-axis system is that the investigations have shown that both the NM and NR algorithms tend to show poorer convergence when manoeuvres are defined in the Earth-axis system, although it would appear to make more physical sense to define the turning circle in the Earth-axis system (Thomson & Bradley, 1998). For the zigzag and pullout manoeuvres, the r (yaw velocity) and Ψ (yaw angle) variables are the quantities considered for the two-output channels because these manoeuvres have most influence on these two-output variables. The input channel is the commanded rudder angle in all cases.

Table 6.2 Convergence of the NM and NR methods without input saturation (Mariner)

Δt, s		20	10	8	6	4	3	2	1	0.5
NM	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	√	√	√	√	√	√	√	√	√
	Pullout	×	√	√	√	√	√	√	√	√
NR	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	√	×	×	×	×	×	×	×	√
	Pullout	×	×	×	×	×	×	×	×	√

Table 6.3 Convergence of the NM and NR methods with input saturation (Mariner)

Δt, **s**		20	10	8	6	4	3	2	1	0.5
NM (not constrained)	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	√	√	√**?**	√**?**	√**?**	√**?**	√	√	√
	Pullout	√	√	√**?**	√**?**	√**?**	√**?**	√	√	√
NM (constrained)	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	√	√	√	√	√	√	√	√	√
	Pullout	√	√	√	√	√	√	√	√	√
NR	Turnc	√	√	√	√	√	×	×	×	×
	Zigzag	√	×	×	×	×	×	×	×	×
	Pullout	×	×	×	×	×	×	×	×	×

Two sets of tests, without or with input saturation, have been carried out and the results are compared with those of the NR methods shown in Table 6.2 and Table 6.3. In these tables the abbreviation "Turnc" represents the turning-circle manoeuvre, the symbol √ stands for convergence, √**?** represents convergence but bad consistency, and × means no convergence. The term "consistency", as used here, relates to the difference between the results from the FFS using calculated inputs and the corresponding values for the desired manoeuvre.

Table 6.2 and Table 6.3 show that even for an ideal input value within the saturation limit, the convergence of the NM method is better than that of the NR approach. In Table 6.2, the NM method achieves good convergence for all three manoeuvres, except for the case of $\Delta t = 20\ s$ for the pullout manoeuvre. There is a cross in this box because the Mariner model fails to generate the ideal pullout manoeuvre without input saturation. It does not, in this particular case, mean non-convergence. The same explanation can be applied to the cross at $\Delta t = 20$ s for the NR method for the pullout manoeuvre. From Table 6.3, it can be

observed that the NR method can only converge for the turning-circle manoeuvre for values of Δt larger than 4 s but fails for intervals below that value. However, the NR method cannot converge for the zigzag and pullout manoeuvres except for the case of $\Delta t = 20$ s for the zigzag.

Some results in Table 6.2 and Table 6.3 are plotted in Fig. 6.10 to Fig. 6.13. The results for the cases without saturation are ignored because they are similar to these plots since the input value to generate the ideal manoeuvre is far smaller than the saturation limit.

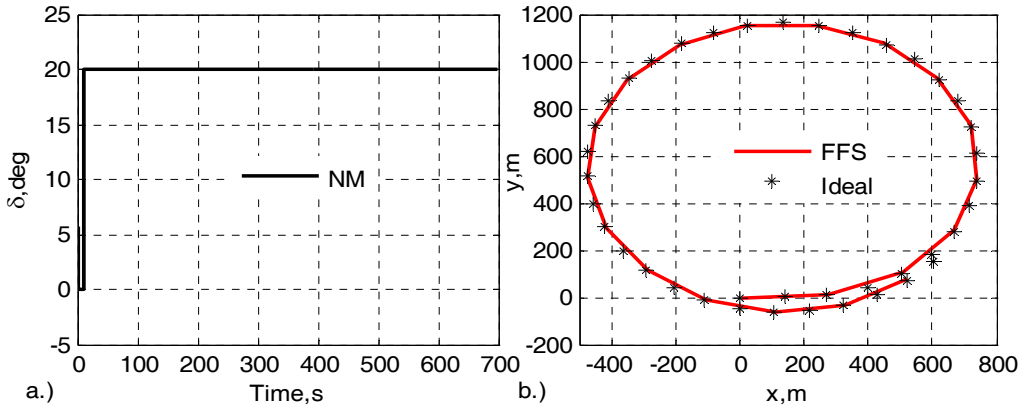

Fig. 6.10 Inverse simulation of the Mariner ship with saturation limits and the corresponding FFS results compared with the ideal manoeuvre (Δt =1 s, turning circle, NM method)

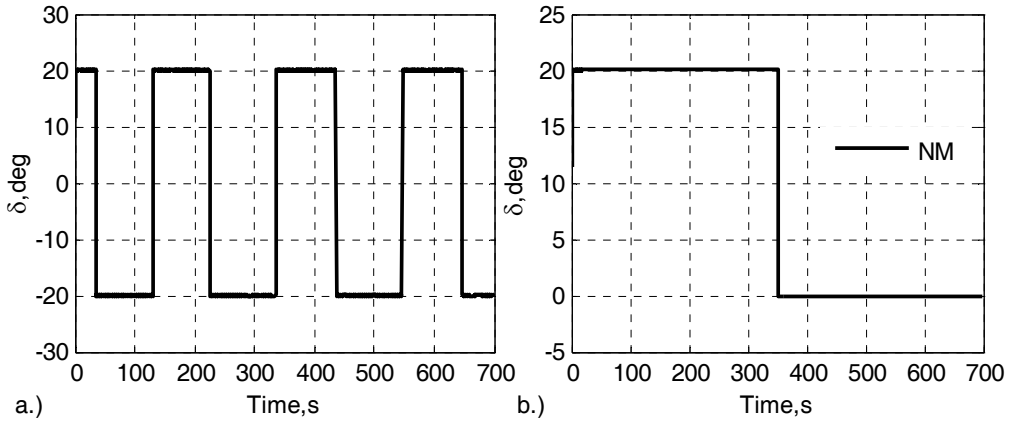

Fig. 6.11 Plots of rudder angle for zigzag (a) and pullout (b) manoeuvres obtained from inverse simulation of the Mariner ship with saturation limits (Δt =1 s, NM method)

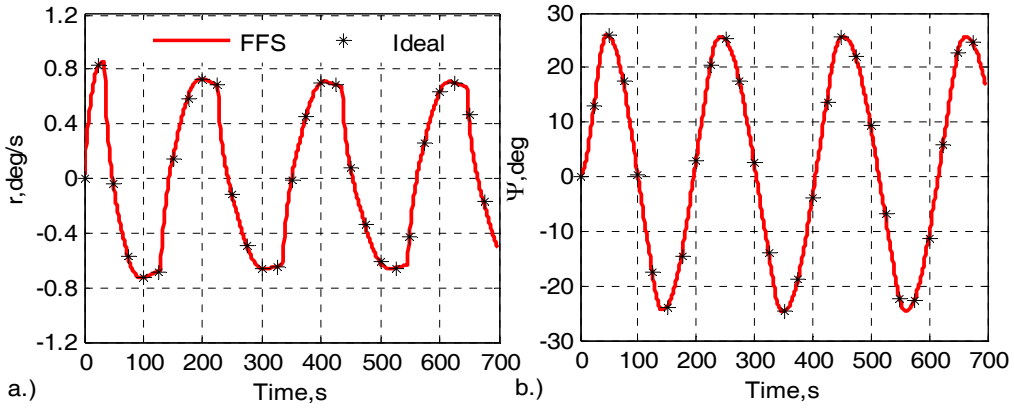

Fig. 6.12 Results obtained from the FFS of the Mariner ship with saturation limits showing comparison with the ideal manoeuvre (Δt =1 s, zigzag, NM method)

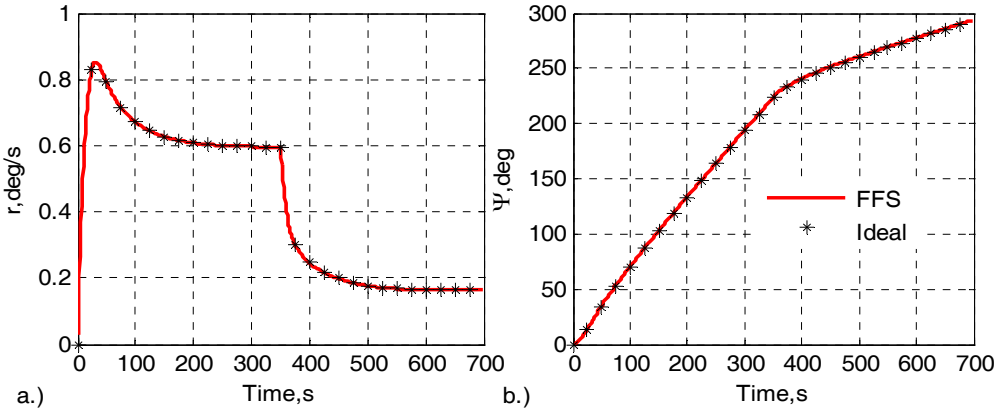

Fig. 6.13 Results obtained from the FFS of the Mariner ship with saturation limits showing comparison with the ideal manoeuvre (Δt =1 s, pullout, NM method)

One of the reasons for different convergence properties for the NR and NM methods without input constraints is due to the existence of a second constrained quantity – the rudder-rate limit. It has been found that during the inverse simulation process the rudder rate may sometimes be above this limit for both these methods. If this saturation effect is not included, the NR method is found to achieve good convergence for all the values of Δt considered, provided a smooth manoeuvre is implemented, such as the turning circle, as shown in Table 6.2 and Fig. 6.10. Moreover, if the value Δt is small enough, the NR method also can show good convergence for severe manoeuvres such as the zigzag (Fig. 6.12) and pullout (Fig. 6.13), as shown in Table 6.2. This is because, when Δt is smaller, the searching space becomes narrower. Therefore, there is an increased probability that the

NR method will reach the solution. However, when the saturation effect is included it can be observed, from Table 6.3, that the NR method still fails for most Δt values. The reason for its convergence at several Δt points, which are large for the turning-circle and zigzag manoeuvres in Table 6.3 as well as in Table 6.2, is that the discretization process inherent in the inverse simulation approach eliminates the information containing the turning points (Fig. 6.12) or transient points (Fig. 6.13). In addition, it is found that when saturation is included the NR method always fails to converge for the pullout manoeuvre around $t = 350$ s where there is a transient. The reasons for this non-convergence have been given in Section 6.2 (Fig. 6.2).

The information about the actual input required by the "Mariner" model to generate the manoeuvre can also be obtained from the inverse simulation process. Fig. 6.10 shows that the actual inputs are the same as the set value for the turning circle after the executed time point $t = 10$ s . The amplitude value of 20 deg in Fig. 6.11a is completely consistent with the given values and the square-pulse shape meets the characteristics of the zigzag manoeuvre. Finally, the step down around $t = 350$ s in Fig. 6.11b is consistent with the transient point in the defined manoeuvres, as shown in Fig. 6.13. All this information can help us to understand the dynamics of the model being considered.

6.4.3 Application to a nonlinear Container ship model

The nonlinear Container ship model involves two inputs – the rudder angle and propeller speed. The parameters configured to generate the manoeuvre are: the time point for rudder execution is 10 s for both the turning circle and the zigzag; the set values for the rudder angle and propeller speed are –35 deg ($\delta_{\min} = -10$ deg) and 80 rpm ($n_{\max} = 160$ rpm), respectively; the cost function or the zigzag and pullout manoeuvres are defined based on Eq. (6.3), and take the form shown in the following equation:

$$L^2[\boldsymbol{u}(t_k)] = \{\dot{\boldsymbol{g}}_1[\boldsymbol{u}(t_k), \boldsymbol{x}(t_{k+1})] - \dot{\boldsymbol{y}}_{d_1}(t_{k+1})\}^2 + \{\dot{\boldsymbol{g}}_2[\boldsymbol{u}(t_k), \boldsymbol{x}(t_{k+1})] - \dot{\boldsymbol{y}}_{d_2}(t_{k+1})\}^2 \qquad (6.12)$$

The first-derivative terms (ideal values) in Eq. (6.12) are obtained from the model simulation beforehand. This approach can avoid the non-smoothness of the zigzag and

pullout manoeuvres. The cost function for the turning circle follows Eq. (6.11). The outputs defined for the Container ship model follow the rules applied for the "Mariner" model. Apart from the three constrained conditions contained in Table 6.1, the shaft acceleration also changes as follows:

$$\begin{cases} n_a > 0.3,\ \dot{n}_a = \dfrac{n_a}{5.65} * (n - n_a) * 60 \\ \text{otherwise},\ \dot{n}_a = \dfrac{1}{18.83} * (n - n_a) * 60 \end{cases} \tag{6.13}$$

where n_a is the actual shaft velocity.

Table 6.4 Convergence of the NM and NR methods without input saturation (Container)

Δt, **s**		10	8	6	4	3	2	1	0.5	0.2
NM	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	×	√	×	√	√	√	√	√	×
	Pullout	×	×	×	√	√	√	×	√	×
NR	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	×	×	×	×	√	√	√	√	×
	Pullout	×	×	×	×	×	×	×	×	×

Table 6.4 shows the convergence qualities of the inverse simulation methods without input saturation. The NM and NR methods achieve similar convergence qualities for the turning circle. This is probably due in part to its smooth property compared with the other two manoeuvres being considered. For the zigzag manoeuvre, the convergence quality of the NM results is slightly better than those from the NR method. However, for the pullout manoeuvre, the NR method fails completely to converge. The main reason for poor convergence for the NR method is the same as for the cased of the "Mariner" ship in that the algorithm cannot overcome the effects of the transient point around $t = 350$ s .

Table 6.5 Convergence of the NM and NR methods with input saturation (Container)

Δt, **s**		20	10	8	6	4	3	2	1	0.5
NM (constrained)	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	×	×	×	×	√	√	√	√	√
	Pullout	√**?**	√**?**	√**?**	√**?**	√	√	√	√	√
NR	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	×	×	×	×	×	×	×	×	×
	Pullout	×	×	×	×	×	×	×	×	×

When the model being considered includes input saturation (Table 6.5), good convergence is still obtained for both the NR and NM methods for the turning-circle manoeuvre. However, the NR method fails completely to converge for both the zigzag and pullout manoeuvres. The convergence of the NM method for these two manoeuvres also becomes worse and its good convergence can only be achieved for Δt values smaller than $\Delta t = 4$ s. This again shows the negative effect of input saturation on the inverse simulation.

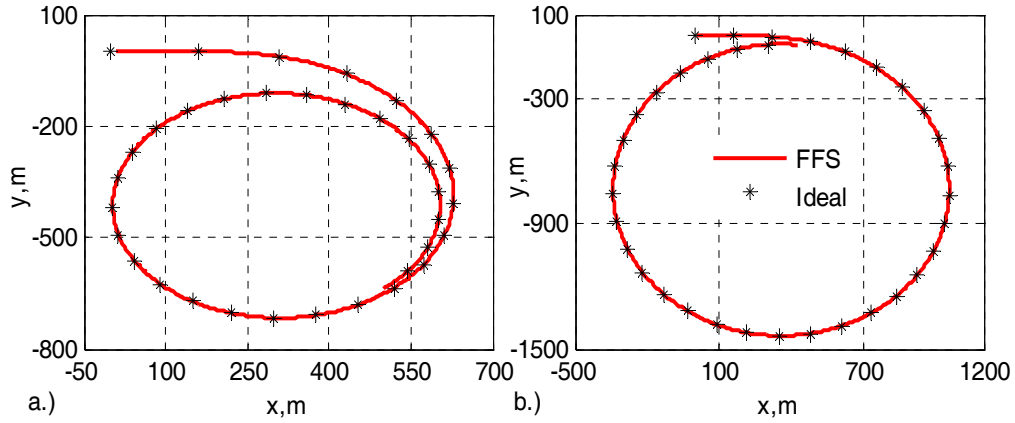

Fig. 6.14 Results obtained from the FFS of the Container ship without (a) and with (b) saturation limits showing comparison with the ideal manoeuvre (Δt = 1 s, turning circle, NM method)

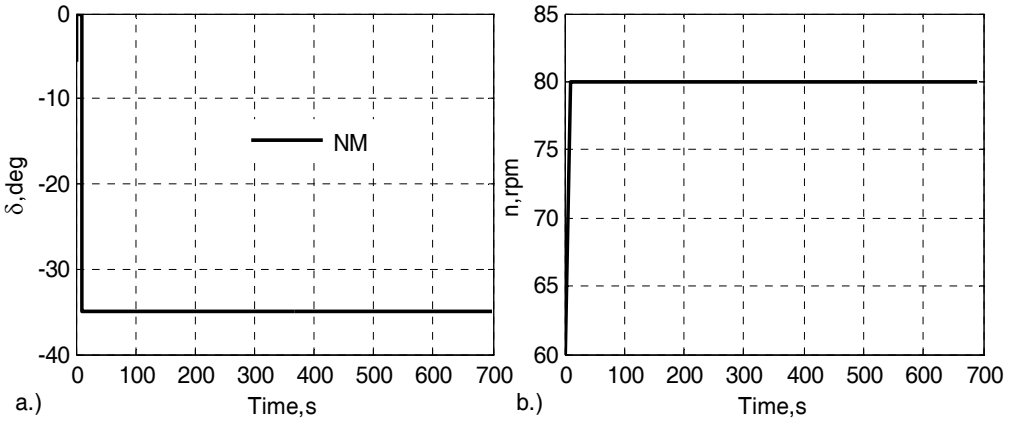

Fig. 6.15 Inputs obtained from inverse simulation of the Container ship without saturation limits (Δt = 1 s, turning circle, NM method)

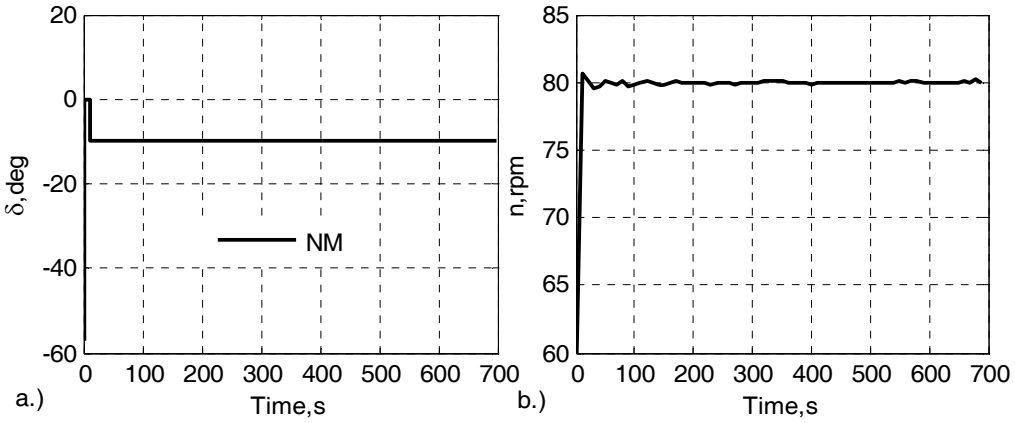

Fig. 6.16 Inputs obtained from inverse simulation of the Container ship with saturation limits (Δt = 1 s, turning circle, NM method)

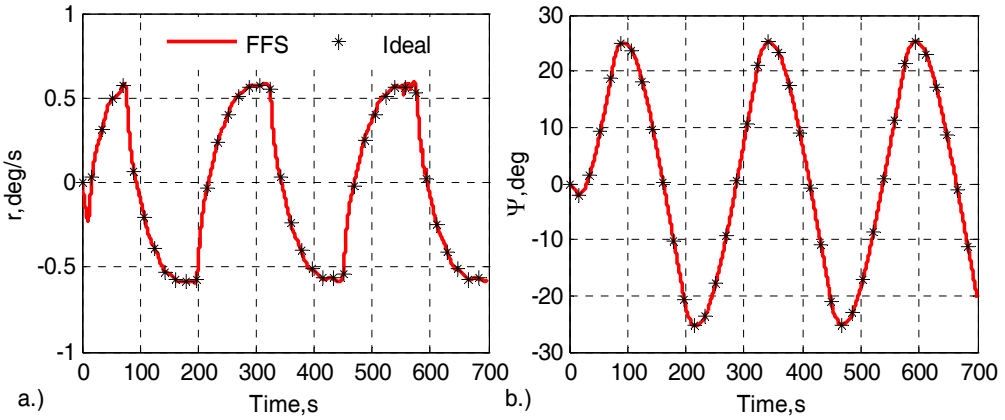

Fig. 6.17 Results obtained from the FFS of the Container ship with saturation limits showing comparison with the ideal manoeuvre (Δt = 0.2 s, zigzag, NM method)

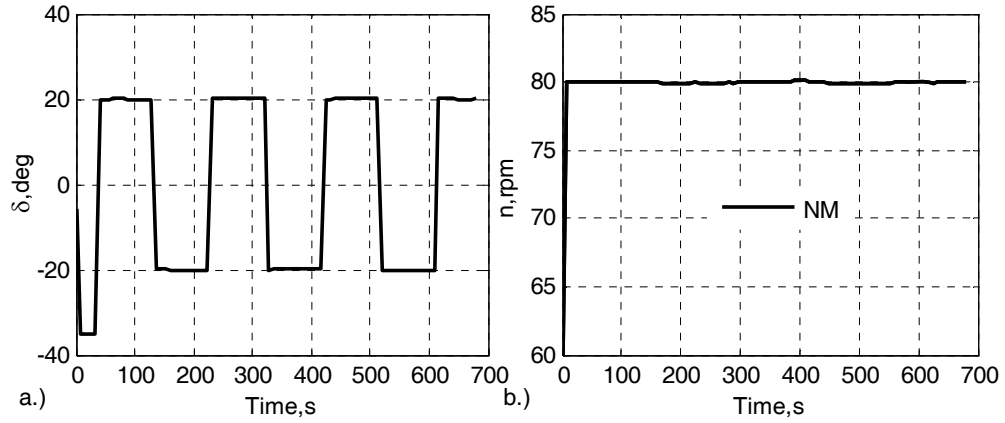

Fig. 6.18 Inputs obtained from inverse simulation of the Container ship without saturation limits (Δt = 1 s, zigzag, NM method)

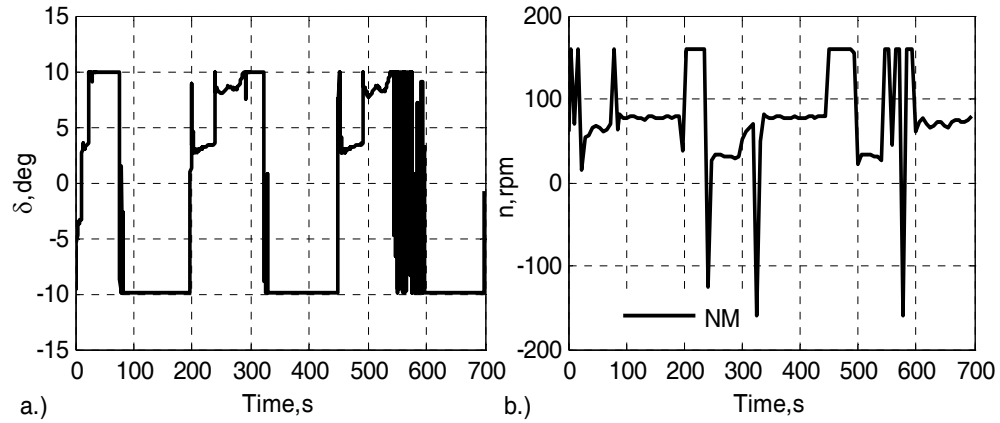

Fig. 6.19 Inputs obtained from inverse simulation of the Container ship with saturation limits (Δt = 0.2 s, zigzag, NM method)

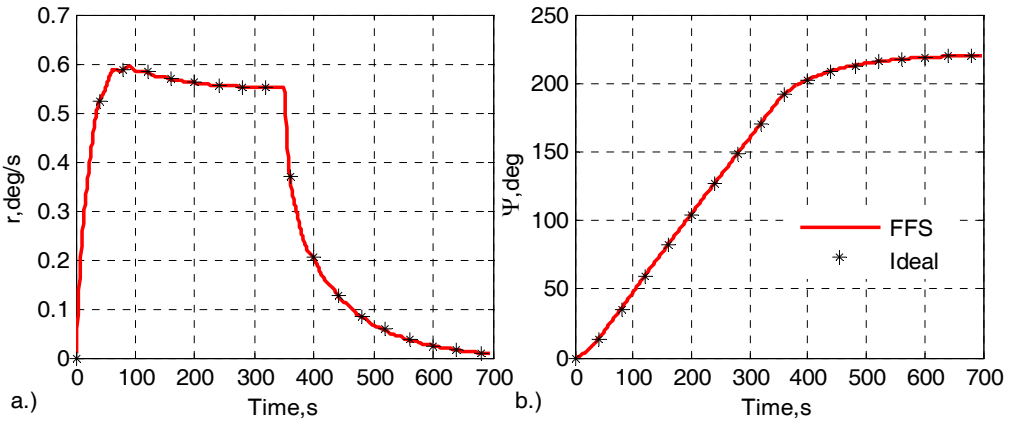

Fig. 6.20 Results obtained from the FFS of the Container ship with saturation limits showing comparison with the ideal manoeuvre (Δt = 1 s, pullout, NM method)

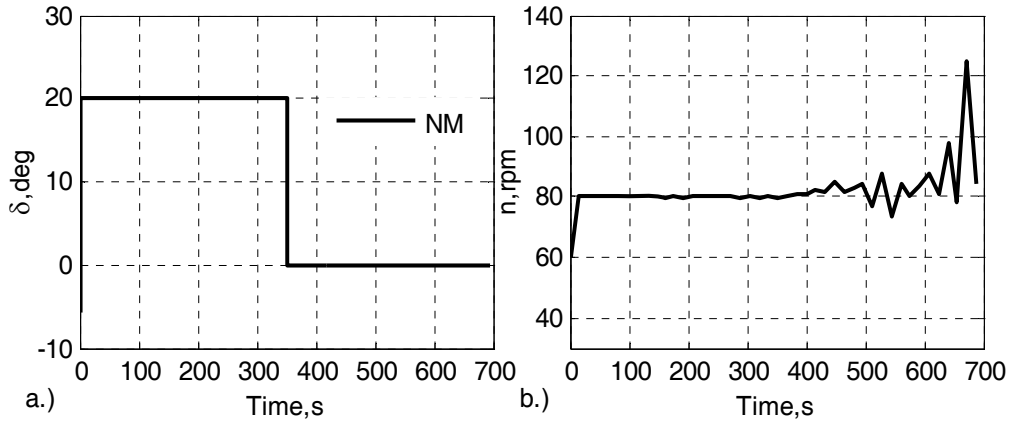

Fig. 6.21 Inputs obtained from inverse simulation of the Container ship without saturation limits (Δt =2 s, pullout, NM method)

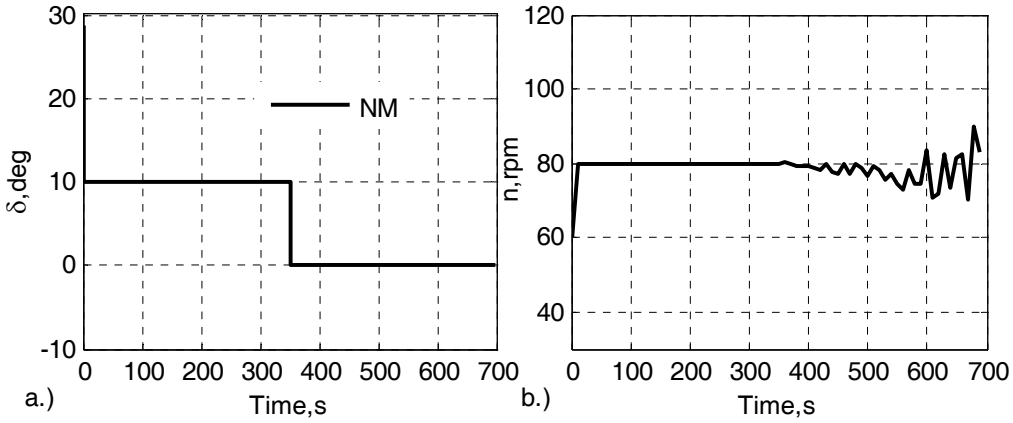

Fig. 6.22 Inputs obtained from inverse simulation of the Container ship with saturation limits (Δt =1 s, pullout, NM method)

Fig. 6.14 to Fig. 6.22 show some results from the above series of tests. Compared with the results for no saturation, inverse simulation with saturation gives a perfect turning circle as shown in Fig. 6.14. Also, the calculated input (δ) in this case is limited to the saturation level of –10 deg in Fig. 6.16 instead of being equal to the set value of –35 deg as applies without saturation (Fig. 6.15(a)). The NM method also achieves good results in Fig. 6.18 for the zigzag without input saturation but not for the situation with saturation, where the results do not match their expected values (10 deg) over the first few seconds at the beginning of each square-wave pattern, as shown in Fig. 6.19a. However, it is interesting to find that although the calculated inputs (δ and n) do not agree well with their ideal values, the results from the FFS with these values still agree well with the ideal manoeuvres. This in fact is a multi-solution phenomenon and has been mentioned previously by Gao and Hess (1993). Furthermore, the results of the n channel in Fig. 6.21 and Fig. 6.22 show oscillations which begin around $t = 350$ s in the later part of the record. The reasons for this have been given already in Section 6.2.

6.4.4 Application to a nonlinear Tanker ship model

In this subsection, the constrained NM method is applied to a nonlinear Tanker model, which involves three inputs - rudder angle (δ), propeller speed (n), and depth of water (h). The parameters configured to generate the manoeuvres are as follows: the time point at

which rudder movement is executed is 10 s for both the turning circle and the zigzag; the set values for these manoeuvres are –20 deg ($\delta_{\max} = 10$ deg), 80 rpm ($n_{\max} = 160$ rpm), and 200 m ($h_{\min} = 18.46$ m), respectively. Hence, the applications in this section represent another kind of redundancy situation in that the number of inputs (three) is larger than the number of outputs (two). The output manoeuvres as well as the cost function are defined by following the rules applied to the Container ship model. The results from the same series of experiments as the previous sections are shown in Table 6.6.

Table 6.6 Convergence of the NM and NR methods with input saturation (Tanker)

Δt, s		20	10	8	6	4	3	2	1	0.5
NM (constrained)	Turnc	√	√	√	√	√	√	√	√	√
	Zigzag	√	√	√**?**	√	√	√**?**	√**?**	√**?**	√**?**
	Pullout	√	√	√**?**	√	√	√	√**?**	√**?**	√**?**
NR	Turnc	×	×	×	×	×	×	×	×	×
	Zigzag	×	×	×	×	×	×	×	×	×
	Pullout	×	×	×	×	×	×	×	×	×

Table 6.6 shows the comparison of the convergence qualities of the NR and NM approaches. Because of the input redundancy and the additional input saturation, the NR method even shows cases of no convergence for the turning-circle manoeuvre. As with the Container ship, it fails to converge for the other two manoeuvres. For the NM method, the convergence quality also decreases for the zigzag and pullout manoeuvre although it still obtains results with good convergence for the turning-circle manoeuvre. This problem is possibly due to the increased complexity of this application of inverse simulation compared with the other ship models. Some results from the above series of tests are shown in Fig. 6.23 to Fig. 6.27 as follows:

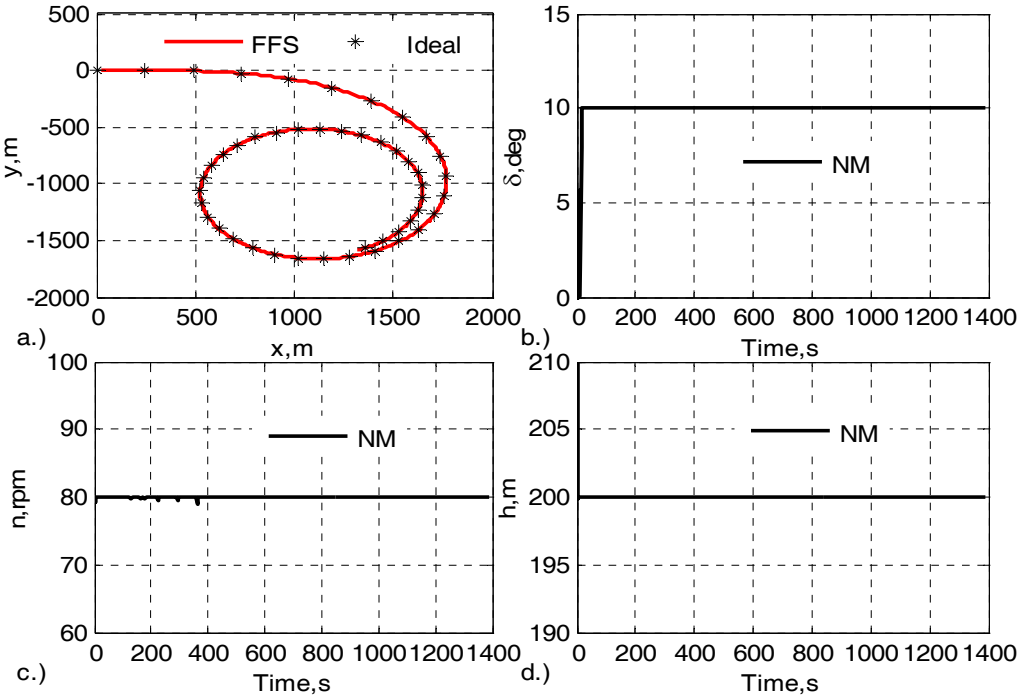

Fig. 6.23 Inverse simulation of the Tanker ship with saturation limits and the corresponding FFS results compared with the ideal manoeuvre (Δt = 3 s, turning circle, NM method)

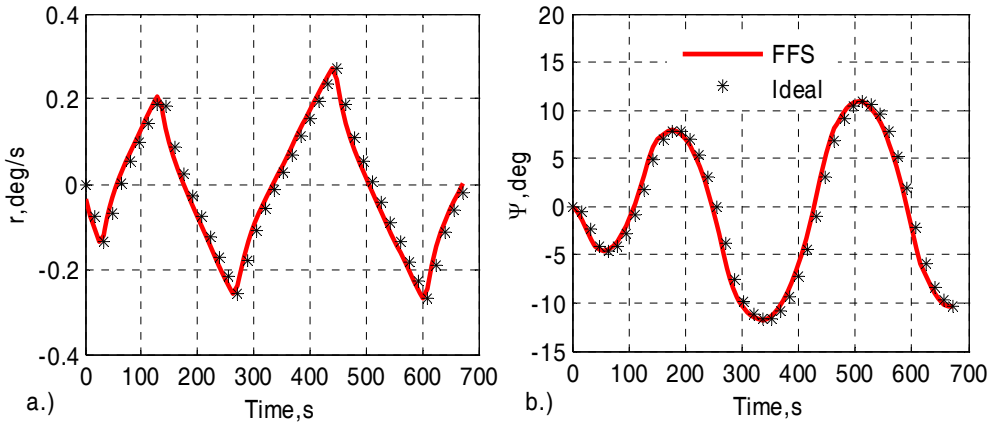

Fig. 6.24 Results obtained from the FFS of the Tanker ship with saturation limits showing comparison with the ideal manoeuvre (Δt = 8 s, zigzag, NM method)

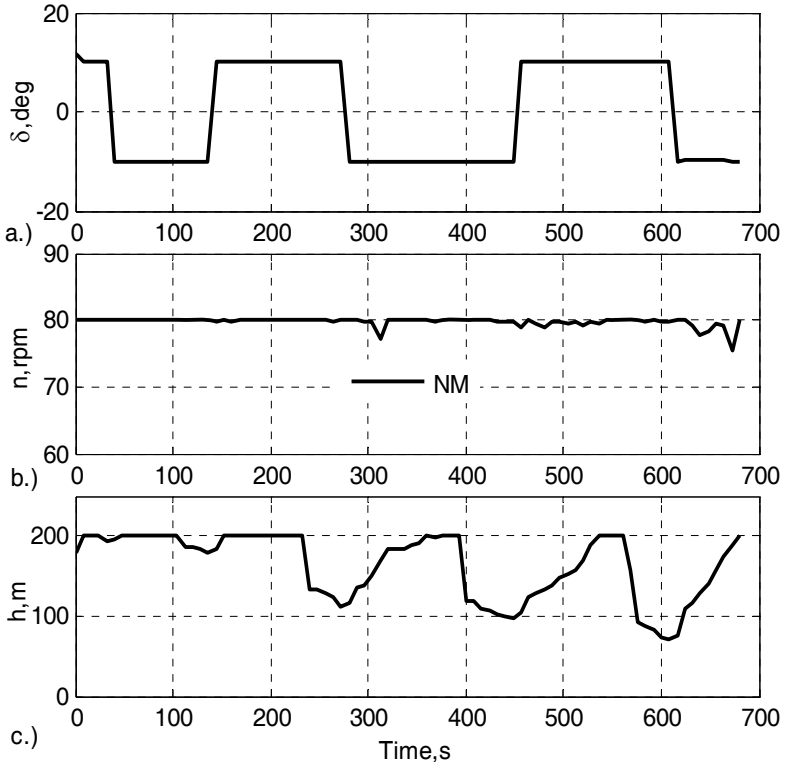

Fig. 6.25 Inputs obtained from inverse simulation of the Tanker ship with saturation limits (Δt = 8 s, zigzag, NM method)

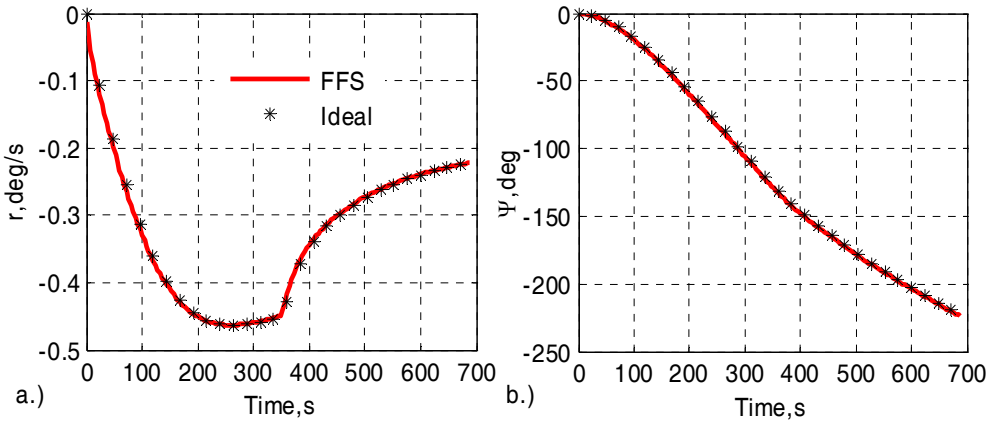

Fig. 6.26 Results obtained from the FFS of the Tanker ship with saturation limits showing comparison with the ideal manoeuvre (Δt = 4 s, pullout, NM method)

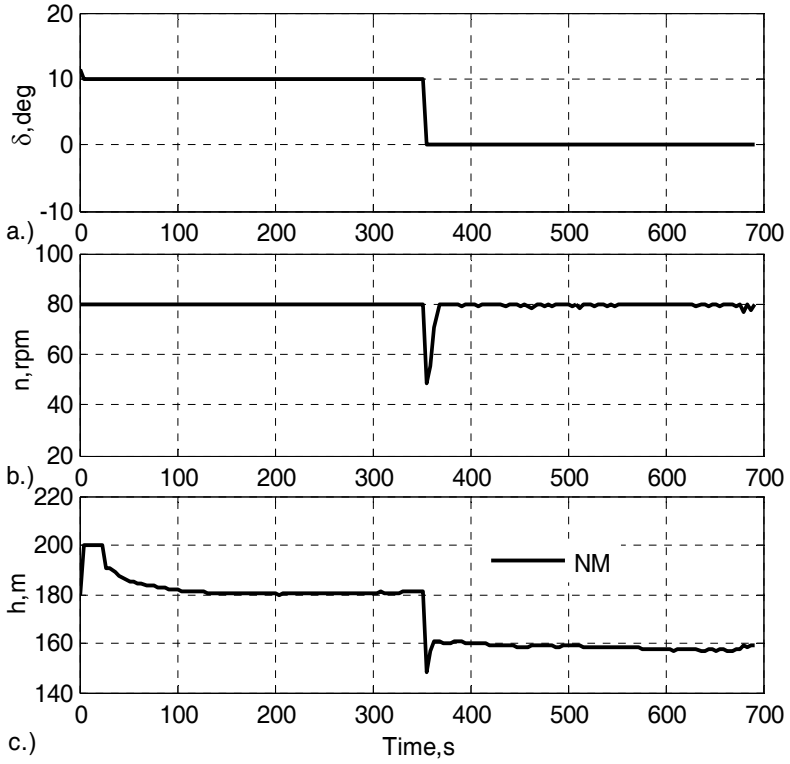

Fig. 6.27 Inputs obtained from inverse simulation of the Tanker ship with saturation limits (Δt = 4 s, pullout, NM method)

The output results calculated from the inverse simulation process on the Tanker model agree well with the ideal manoeuvres as shown in Fig. 6.23a. The two input channels, shaft velocity, and depth, agree well with the set values. The third channel is also limited to the saturation level of 10 deg. However, the situation becomes slightly worse for the zigzag and pullout cases, as shown in Fig. 6.24 to Fig. 6.27. For instance, the depth results in Fig. 6.25c and Fig. 6.27c for both manoeuvres do not agree with the expected values (160 rpm). The other two input channels are consistent with the saturation limits except for one pulse in the shaft velocity channel both for the pullout manoeuvre and the zigzag manoeuvre, along with the step-down points. However, the outputs from the FFS using the calculated three inputs still comply with the ideal manoeuvres, as shown in Fig. 6.24 and Fig. 6.26.

6.4.5 Application to a nonlinear AUV model

In this subsection, the constrained NM method is applied to a nonlinear AUV model, which involves six inputs – rudder angle (δ_r), port and starboard stern plane (δ_s), top and bottom bow plane (δ_b), port bow plane (δ_{bp}), starboard bow plane (δ_{bs}), and propeller shaft speed (n), and also four outputs – positions in x, y and z-directions and yaw angle (Ψ). The parameters configured to generate the manoeuvres are as follows: the time point at which rudder movement is executed is 5 s for both the turning circle and the zigzag. Hence, the application discussed in this section involves a redundant situation in that the number of inputs (six) is larger than the number of outputs (four). The cost function is defined by Eq. (6.1) with dimension equal to four.

Table 6.7 Input values to generate the ideal trajectory (AUV)

Type	δ_r, deg	δ_s, deg	δ_b, deg	δ_{bp}, deg	δ_{bs}, deg	n, rpm
Turnc	25	0	0	0	0	1200
Zigzag	15	15	15	15	15	1200

Table 6.8 Convergence of the NM and NR methods with input saturation (AUV)

Δt, **s**		8	7.5	7	6.5	6	5	4	3	2
NM (constrained)	Turnc	×	√	√	√**?**	√	√**?**	√	√**?**	×
	Zigzag	×	√	√	√	√	√	√	×	×
NR	Turnc	×	×	×	×	×	×	×	×	×
	Zigzag	×	×	×	×	×	×	×	×	×

The set values for these manoeuvres are shown in Table 6.7. There the rudder angle for the turning-circle manoeuvre exceeds the saturation level. Therefore, the actual rudder input for the AUV model is the saturation value (20 deg). The input values configured for the zigzag manoeuvre are all within the limits. The results of inverse simulation of the NM and NR methods from these manoeuvres on the AUV model with input saturation, as shown in Table 6.8, are completely different. The NM method achieves good convergence both for the turning-circle and zigzag manoeuvres. In contrast, the NR method fails to converge for all situations even for the case of the smooth turning-circle manoeuvre. Besides, it also fails for situations without input saturation. The reason for this may be due to the failure of the

NR method to deal with such a complicated model including the input constraints. For better analysis, some results from the above tests are plotted out as shown in the following.

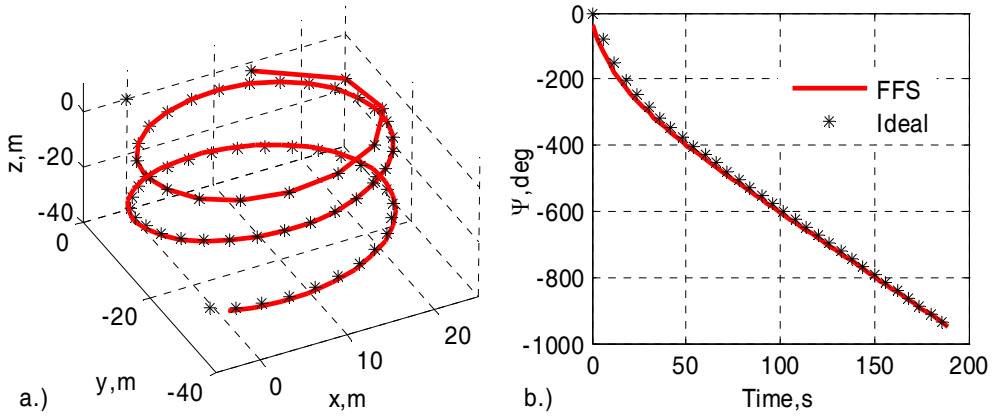

Fig. 6.28 Results obtained from the FFS of the AUV ship with saturation limits showing comparison with the ideal manoeuvre (Δt = 3 s, turning circle, NM method)

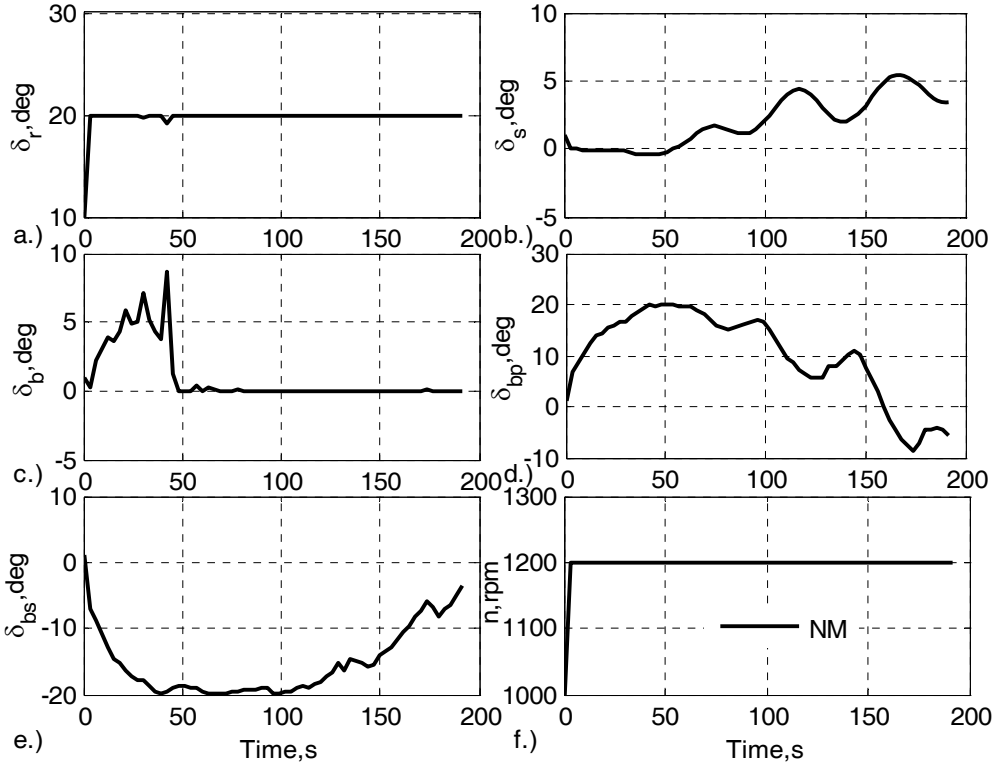

Fig. 6.29 Inputs obtained from inverse simulation of the AUV with saturation limits (Δt = 3 s, turning circle, NM method)

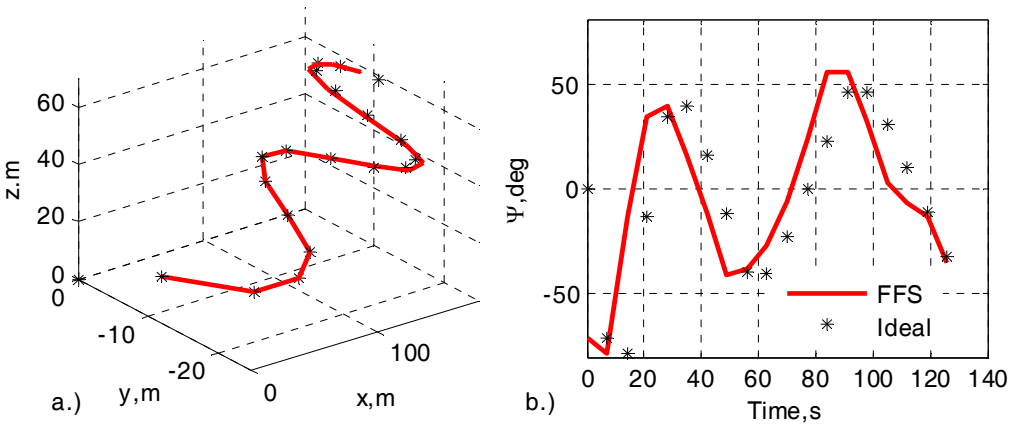

Fig. 6.30 Results obtained from the FFS of the AUV ship with saturation limits showing comparison with the ideal manoeuvre (Δt = 7 s, zigzag, NM method)

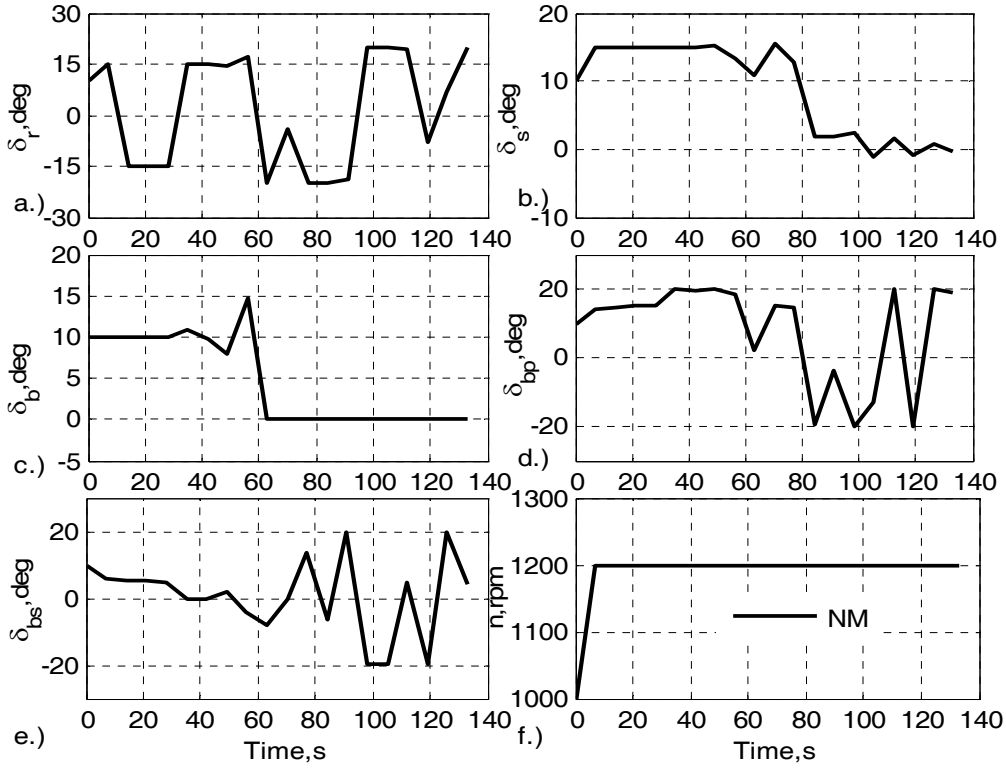

Fig. 6.31 Inputs obtained from inverse simulation of the AUV with saturation limits (Δt = 7 s, zigzag, NM method)

Fig. 6.28 and Fig. 6.30 show that the output results calculated from the FFS with the calculated inputs applied to the AUV model agree well with the ideal manoeuvres except for the case of the yaw angle Ψ which slightly diverges. In Fig. 6.29, the calculated input

values – δ_r, δ_s, δ_b, and n, comply well with the expected values. The results for the channels δ_{bp} and δ_{bs} differ from the ideal values but are within the saturation limits. However, the outputs from the FFS using these calculated six inputs are still consistent with the ideal manoeuvres. This is again a multi-solution phenomenon and the same phenomenon also appears in Fig. 6.31. The outputs of the FFS with these calculated inputs still comply well with the ideal manoeuvres. This shows that the control efforts required to perform such a manoeuvre are not unique. Therefore, inverse simulation possibly provides a tool for control allocation (Boskovic & Mehra, 2002) or facilitates finding an optimal trajectory from the available data (Williams, 2005). The slight divergence in the yaw angle channel may arise from the relatively large Δt value rather than from the poor input consistency since the other three output channels x, y, and z follow the ideal values.

6.5 Summary

A new, completely derivative-free, procedure has been developed in this chapter for inverse simulation, based on the constrained NM algorithm. The problems of inverse simulation associated with input saturation and discontinuous manoeuvres have been explored and discussed. The proposed approach avoids the augmented *Lagrangian* method to solve the constrained conditions by one-step forward simulation and the application of input transformations.

Simulations of five nonlinear marine vehicle models have been considered. These cases represent three different situations in terms of the number of inputs and outputs. Manoeuvres investigated includes the one generated using a third-order reference model, the turning-circle manoeuvre, a zigzag type of manoeuvre, and a pullout manoeuvre. The results show that the new method of inverse simulation provides better convergence and numerical stability for cases involving input saturation or discontinuous manoeuvres. However, for severe manoeuvres such as the zigzag and complex models such as the AUV, a multi-solution phenomenon may appear in the results.

It is suggested that the multi-solution phenomenon has potential advantages in dealing with control reallocation and may allow the optimal control effort to be found by modification of the cost-function definition. In addition, the NM method can form a useful reference method that can allow a better understanding of some numerical problems associated with the other commonly used methods. Also the knowledge gained from inverse simulation using the NM approach can help in the design of a FFC, as will be shown in the later chapters.

Chapter 7

Feedback Controller Design

Contents

This chapter focuses on design issues associated with the FBC in the 2DOF scheme for the Norrbin ship model, for a linear Lynx-like helicopter model, and for the Son and Nomoto full nonlinear Container ship model by the mixed-sensitivity (K/KS) H_∞ control methodology. The performance of the controllers designed using this approach will be analysed and investigated in detail by means of applications involving the different systems mentioned above in a variety of situations.

7.1 Introduction

The 2DOF scheme has been reviewed in detail in Chapter 2. As mentioned in that chapter, this scheme consists of two parts – the FFC and the FBC, each of which has a different function. The FFC is believed to be able to improve the tracking performance and the FBC can increase robust stability and achieve good disturbance rejection. In addition, the separate procedure to design the FFC and FBC (Giusto & Paganini, 1999), as introduced in

Chapter 2, has been adopted in this thesis. Therefore, it is possible to design a FBC separately for the 2DOF scheme provided that a suitable design methodology can be found.

In this chapter, the mixed-sensitivity H_∞ control methodology is adopted to design the FBC. The results achieved here will pave the way for investigating the performance of the 2DOF system by introducing the FFC in the next chapter. This chapter will begin with a review the historical development of the H_∞ control algorithm as well as presenting the latest contributions made to this field. Then, the H_∞ control algorithm is implemented to design the FBC for the Norrbin model, for a linear Lynx-like helicopter model, and for the Son and Nomoto full nonlinear Container ship model. In addition, the results from the application on the nonlinear Container ship model will be compared with the results obtained from application of the classical LQ method. Finally, conclusions are presented based on these discussions and results found from the applications.

7.2 Review of the H_∞ control algorithm

Since the pioneering work of Zames (1981), the H_∞ algorithm has been a dominant method of design in the multivariable control field and has been investigated intensively in various fields of application. Therefore, only a brief introduction to some of the important aspects of the H_∞ algorithm is presented in this section. H_∞ control has been recognized to be good at simultaneously meeting the high-precision tracking-performance requirements, reducing the risk of control energy saturation, and improving stability robustness with additive and multiplicative uncertainties (Yang *et al*., 2005). The latest work relating to this control algorithm in the field of aircraft applications can be found in the references (La Civita *et al*., 2003; Luo *et al*., 2003; Moghaddam & Moosavi, 2005; Postlethwaite *et al*., 2005; Tanner & Geering, 2003) and in the marine field application in the references (Hu *et al*., 2003; Wang & Ren, 2004).

In H_∞ controller design, the order of the compensator represents a major challenge. It is determined by the system dynamics plus all of the additional state variables associated with

the weighting functions at the plant input and outputs. Thus, the order of the resulting compensator may be too large to be readily implemented. In fact, it is a technique that achieves desired levels of performance robustness but at the cost of increasing the controller dimension (Byrns, Jr & Calise, 1994). Some authors have presented methods to reduce the complexity and order of the H_∞ controller. Osborne (1994) applied an order-reduction algorithm to design a controller with the standard H_∞ algorithm. The designed controller is proper instead of strictly proper. Byrns *et al* (1994) presented an approach that designs a fixed-order H_∞ compensator by using an observer canonical form to represent the dynamic compensator and by uniquely selecting the quadratic performance index weighting matrices.

In the implementation of the H_∞ algorithm, all goals are achieved by selecting different kinds of weighting functions since their formulae represent the performance specifications. Hence, the key point within the H_∞ control design process is the choice of proper weighting functions. Information about the methods of weighting-function selection can be found in references such as (Luo *et al*., 2003; Min *et al*., 2005; Yang, Ju & Liu, 1994; Yang, Tai & Lee, 1994; Yang *et al*., 2005). Yang *et al* (2005) introduced the algorithm to select the optimal weighting function using the Taguchi approach and genetic algorithm methods.

The above-mentioned methods can be used to design H_∞ controllers for linear systems. However, these linear controllers have suffered from some drawbacks such as they can be made robust in terms of mathematical meaning but that they are not intrinsically more robust and their valid domains are limited. Recently, the robust H_∞ control problem for nonlinear systems has attracted considerable interest in the aircraft field (Li, 1997; Yang, Liu & Kung, 2002). This technique up to now is still under development. A nonlinear controller can naturally expand the valid region to cover the whole state space instead of the local region within which the traditional linear controller applies. The linear controller from the traditional H_∞ techniques is obtained by solving a filter-type Riccati equation, a control-type Riccati equation, and a coupling condition. In contrast, the nonlinear controller is computed by solving a positive definite function from a nonlinear Hamilton-Jacobi partial differential inequality (Li, 1997).

7.3 Design of a FBC for the Norrbin ship model

The introductory section in Chapter 2 has shown the need to introduce the FBC to overcome the effects of external disturbances and plant uncertainties for the 2DOF control structure. In this section, the mixed-sensitivity H_∞ algorithm has been implemented to design the FBC for the Norrbin ship model (Appendix-D). Unar (1999) designed a FBC for the RZ ship using a multilayer perceptron network trained using the back-propagation learning algorithm. The controller designed in that way showed good robust performance for a number of different operation points (different forward speeds). In contrast, this section will focus on designing the FBC using the *K*/*KS* H_∞ algorithm. Investigations are again based on the RZ ship model but include the effects of disturbances and measurement noise, which were both ignored in Unar's work. The detailed information about the *K*/*KS* algorithm can be found in the book (Skogestad & Postlethwaite, 1996) and its latest applications in other references such as (Fales & Kelkar, 2005; Ortega & Rubio, 2004).

The FBC is designed using the model linearised from Eq. (D.4) around $U = 10$ m/s by ignoring the third equation. The linear model then has the state-space form shown in Eq. (7.1).

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & -\dfrac{d_1}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{m} \end{bmatrix} \delta \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{7.1}$$

An additional disturbance model has been included in the design process and the whole control system structure for this algorithm is as shown in Fig. 7.1

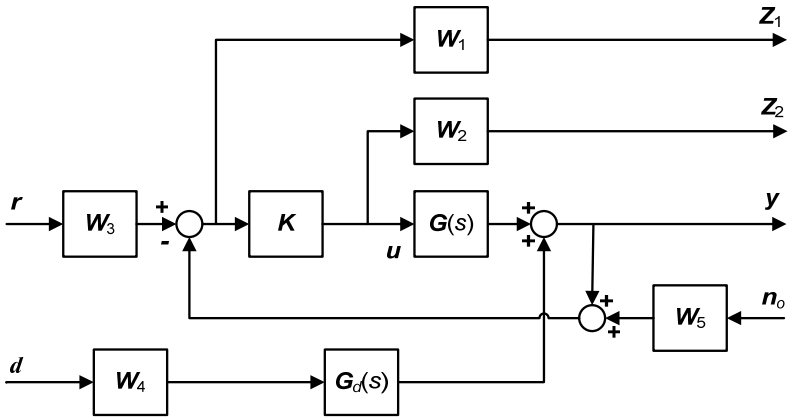

Fig. 7.1 Diagram showing the *K*/*KS* H_∞ control structure

In Fig. 7.1, $\boldsymbol{n}_0$ represents external measurement noise, $\boldsymbol{r}$ stands for the reference state variables, and $\boldsymbol{d}$ is the external disturbance. $\boldsymbol{G}(s)$ and $\boldsymbol{G}_d(s)$ are the nominal model and the disturbance model, respectively. $\boldsymbol{W}_{1\text{-}5}$ represent the weighting functions. The variables $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$, represent the tracking-error signal reshaped by the relevant weighting function and the reshaped control output $\boldsymbol{u}$, respectively. The vector $\boldsymbol{y}$ is the vector of system outputs.

The general configuration for synthesis of the system of Fig. 7.1 is shown by the structure of Fig. 7.2.

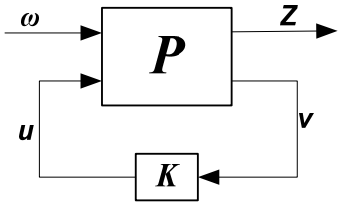

Fig. 7.2 General control configuration

In Fig. 7.2, $\boldsymbol{P}$ is the generalised plant. The vectors $\boldsymbol{\omega}$ and $\boldsymbol{Z}$ are defined as follows:

$$\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{d} \end{bmatrix} \qquad \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}_1 \\ \boldsymbol{Z}_2 \end{bmatrix} \tag{7.2}$$

The essence of the mixed-sensitivity control is to find a controller to minimise the H_∞ norm of the following lower linear fractional transformation (LFT) for $\boldsymbol{\omega}$ to $\boldsymbol{Z}$:

$$\boldsymbol{F}_l(\boldsymbol{P},\boldsymbol{K}) = \left\| \begin{bmatrix} \boldsymbol{W}_1\boldsymbol{S}\boldsymbol{W}_3 & -\boldsymbol{W}_1\boldsymbol{S}\boldsymbol{G}_d\boldsymbol{W}_4 \\ \boldsymbol{W}_1\boldsymbol{K}\boldsymbol{S}\boldsymbol{W}_3 & -\boldsymbol{W}_2\boldsymbol{S}\boldsymbol{G}_d\boldsymbol{W}_4 \end{bmatrix} \right\|_\infty \tag{7.3}$$

where the sensitivity function $\boldsymbol{S}$ is equal to $(\boldsymbol{I}+\boldsymbol{G}\boldsymbol{K})^{-1}$ and $\boldsymbol{W}_5$ is ignored because it is selected as a unit matrix with the corresponding dimension.

As mentioned in the above section, all goals are achieved by selecting different kinds of weighting functions in the application of the H_∞ design algorithm. In Fig. 7.1, the weighting function $\boldsymbol{W}_1(s)$ is used to shape the tracking error between the output signals and the reference input signals. Usually this is selected to be a low-pass filter with a bandwidth equal to that of the disturbance since the disturbance is usually a low-frequency signal. $\boldsymbol{W}_2(s)$ is chosen to be a high-pass filter with a crossover frequency approximately equal to that of the desired closed-loop bandwidth. Through this choice, saturation problems in the actuators can be avoided. $\boldsymbol{W}_3(s)$ is the weighting function on the reference input and the values of its elements depend individually on the priority given to each input for the application being considered. The weighting function $\boldsymbol{W}_4(s) = \alpha\boldsymbol{I}$ (where α is used to adjust the performance against disturbances).

The disturbance model $\boldsymbol{G}_d$ consists of two parts as shown in the diagram of Fig. 7.3.

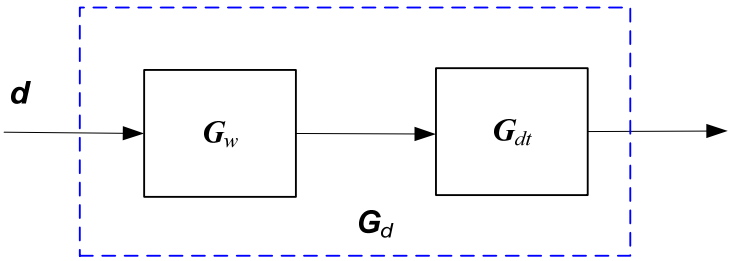

Fig. 7.3 The structure of the disturbance model

In Fig. 7.3, $\boldsymbol{G}_w$ is the linear model used to generate the wave moments and forces and $\boldsymbol{G}_{dt}$ is the model to describe the influence of $\boldsymbol{G}_w$ on the nominal model $\boldsymbol{G}$ as represented in Eq. (7.1). They have the following state-space forms:

$$\boldsymbol{G}_w: \begin{cases} \dot{\boldsymbol{x}}_w = \boldsymbol{A}_w \boldsymbol{x}_w + \boldsymbol{B}_w \boldsymbol{d} \\ \boldsymbol{y}_w = \boldsymbol{C}_w \boldsymbol{x}_w + \boldsymbol{D}_w \boldsymbol{d} \end{cases} \quad \boldsymbol{G}_{dt}: \begin{cases} \dot{\boldsymbol{x}}_{dt} = \boldsymbol{A}_{dt} \boldsymbol{x}_{dt} + \boldsymbol{B}_{dt} \boldsymbol{y}_w \\ \boldsymbol{y}_{dt} = \boldsymbol{C}_{dt} \boldsymbol{x}_{dt} + \boldsymbol{D}_{dt} \boldsymbol{y}_w \end{cases} \tag{7.4}$$

where $\boldsymbol{A}_{dt}$ is equal to the system matrix $\boldsymbol{A}$ in Eq. (7.1) and the selection of $\boldsymbol{B}_{dt}$ depends on the dynamic parameters of the nonlinear RZ model being investigated. The following are the chosen values for the weighting functions $\boldsymbol{W}_{1\text{-}5}(s)$:

$$\boldsymbol{W}_1 = 1.6\frac{s+1.4}{s+0.001} \tag{7.5}$$

$$\boldsymbol{W}_2 = 0.01\frac{s+0.001}{s+0.9} \tag{7.6}$$

$$\boldsymbol{W}_i (i = 3,4,5) = 1 \tag{7.7}$$

The bandwidths of the weighting functions $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are selected to be around 1 rad/s by referring to the bandwidth of the disturbance model $\boldsymbol{G}_d$ (1 rad/s), as well as the rudder control and rate constraints. The inclusion of the integral action in $\boldsymbol{W}_1$ is used to compensate for the wave drift (2nd-order-wave-induced motion). The final cost function value (γ) of the H_∞ norm is 2.05, which is well within the usually acceptable range of 0 to 4 (Skogestad & Postlethwaite, 1996). Better controllers may be found by selecting new sets of weighting functions (Ortega & Rubio, 2004; Yang *et al*., 1994).

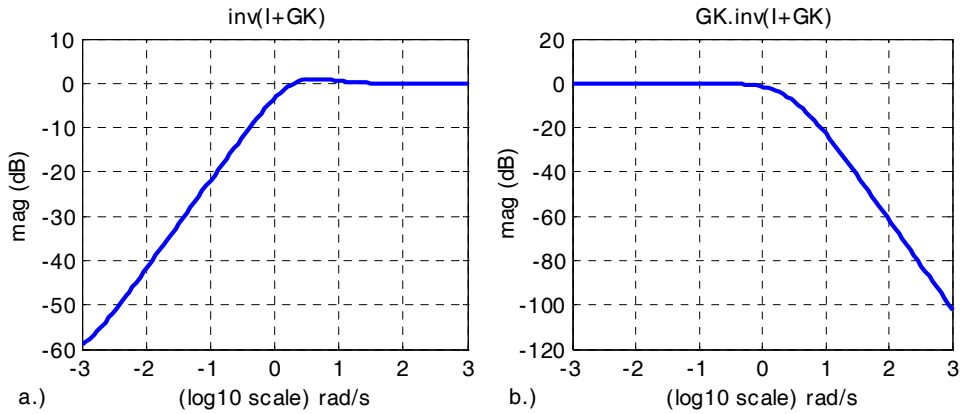

Fig. 7.4 The plots of $(\mathbf{I}+\boldsymbol{GK})^{-1}$ (a) and $\boldsymbol{GK}(\mathbf{I}+\boldsymbol{GK})^{-1}$ (b)

Fig. 7.4a shows a plot of the singular value of the sensitivity function ($\boldsymbol{S}$), which determines the performance in terms of disturbance rejection in the low-frequency range (Skogestad &

Postlethwaite, 1996). Its singular value decreases rapidly in the frequency range of 0 to 1 rad/s, which is consistent with the bandwidth of the disturbance model. The small peak existing in Fig. 7.4a is a NMP phenomenon. Fig. 7.4b shows the singular-value plot of the complementary sensitivity function (***T***), which is useful for determining the tracking performance. In that figure, the singular value of the heading channel (Ψ) has a value of one within the low-frequency band, which suggests that good tracking performance may be achieved for this channel.

In validating the performance of the designed controller, the nonlinear model combined with the steering machine structure described in Fig. 6.4 is considered as the simulation benchmark, as shown in Fig. 7.5:

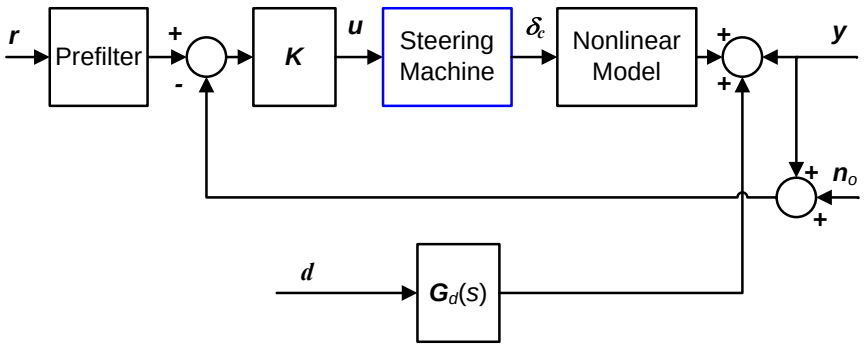

Fig. 7.5 The whole simulation benchmark

The results from a series of simulation runs are shown in Figs. 7.6 to 7.9:

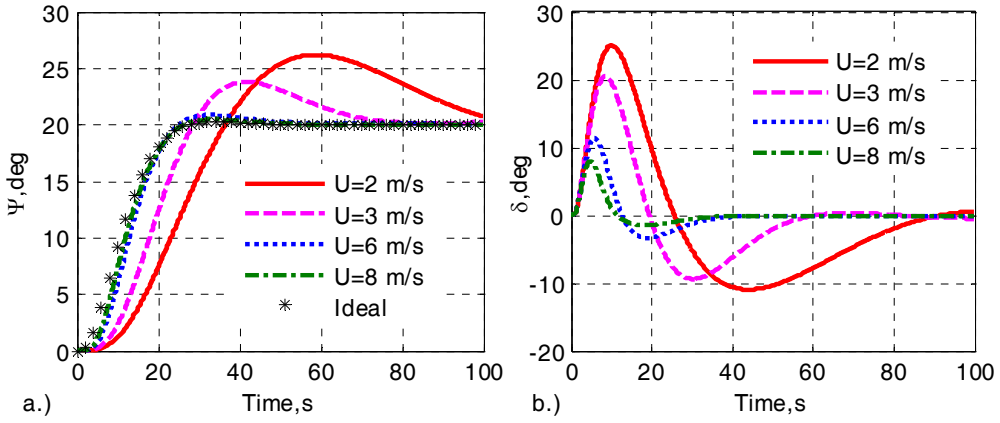

Fig. 7.6 Simulations of the RZ ship with the FBC alone (Ψ_d = 20 deg)

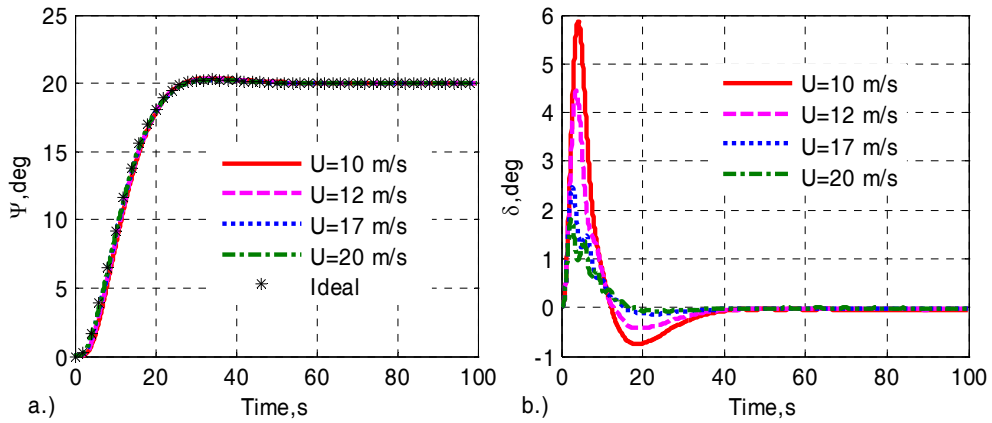

Fig. 7.7 Simulations of the RZ ship with the FBC alone (Ψ_d = 20 deg)

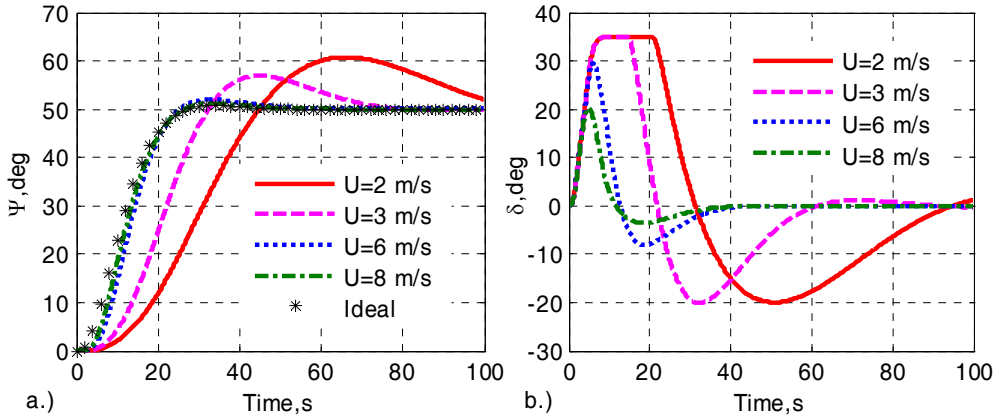

Fig. 7.8 Simulations of the RZ ship with the FBC alone (Ψ_d = 50 deg)

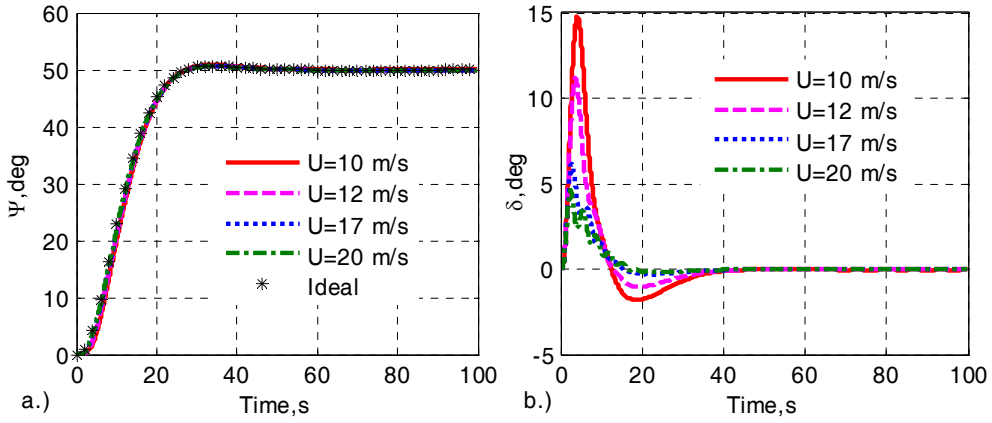

Fig. 7.9 Simulations of the RZ ship with the FBC alone (Ψ_d = 50 deg)

Figs. 7.6 to 7.9 show that the control system represented in Fig. 7.5 achieves good tracking performance as well as perfect rejection of the external disturbance and insensitivity to measurement noise, for ideal heading angles of both 20 deg and 50 deg when the forward speed U is larger than 6 m/s. However, the FBC displays poor tracking performance for $U = 2$ m/s and $U = 3$ m/s. These findings are consistent with the results presented by Unar (1999) and correspond to significant variation of the coefficients of Eq. (D.3). This is obvious from Table D.1 where it may be seen that when the speed decreases from 6 m/s to 2 m/s, the parameter m changes from 43 to 387, $\boldsymbol{d}_1$ from 1.7 to 5, and $\boldsymbol{d}_3$ from 0.46 to 12. Therefore the FBC fails to deal with such severe variations. Compared with these variations at speeds U below 6 m/s, the changes are quite gentle from 6 m/s to 20 m/s, especially from 10 m/s to 20 m/s. These small variations for the larger values of forward speed contribute to a reduction in the control effort required, as shown in the above figures.

For the case of the heading angle of 20 deg, Fig. 7.6 illustrates that the required control efforts are within the saturation level for $U = 3$ m/s and $U = 2$ m/s, in contrast to the results given by the inverse simulation process, as shown in Figs. 6.6 and 6.8. This reduced control effort proves the effectiveness of the FBC system, although saturation limits are still being exceeded for the case of Ψ_d = 50 deg. The latter effect is due to the fact that the larger control effort is required for the larger tracking heading angle and the larger variations of coefficient values for the lower speeds as discussed above.

The FBC designed in this section achieves the required tracking performance and robustness against variation of the forward speed and thus provides an overall level of performance that is as good as the performance obtained by Unar (1999) even allowing for the fact that the current work involves the disturbance model and measurement noise. Moreover, the design approach implemented in this section has some advantages over the approach based on multilayer perceptron networks since it avoids the training procedure which will take quite a long time to be completed (around 31 minutes required for this case). However, a slightly oscillatory phenomenon has appeared in the time history of rudder angle deflections as shown in Figs. 7.7 and 7.9 during the first few seconds of the record for the forward speed U larger than 17 m/s. Analysis has indicated that this problem results

from the large value of time constant τ shown in Fig. 6.4 and adopted in the steering machine model used by Fossen (1994). In the above applications and the work of Fossen, the default value for τ is 1 s, which can be derived from the third equation of Eq. (D.3). In fact, the parameter τ plays the same role as a first-order lag between δ_c and δ, as shown in Eq. (7.8), given that the rudder limiter has not been reached.

$$\delta(s) = \frac{\delta_c(s)}{1+\tau s} \tag{7.8}$$

By reducing the value τ the problem of oscillations can be eliminated, as shown by the following results:

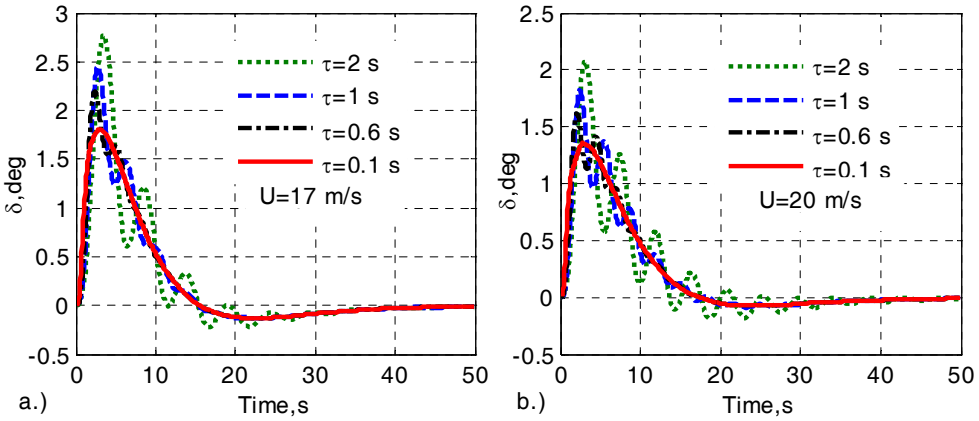

Fig. 7.10 Rudder angles for a range of different τ values (Ψ_d = 20 deg)

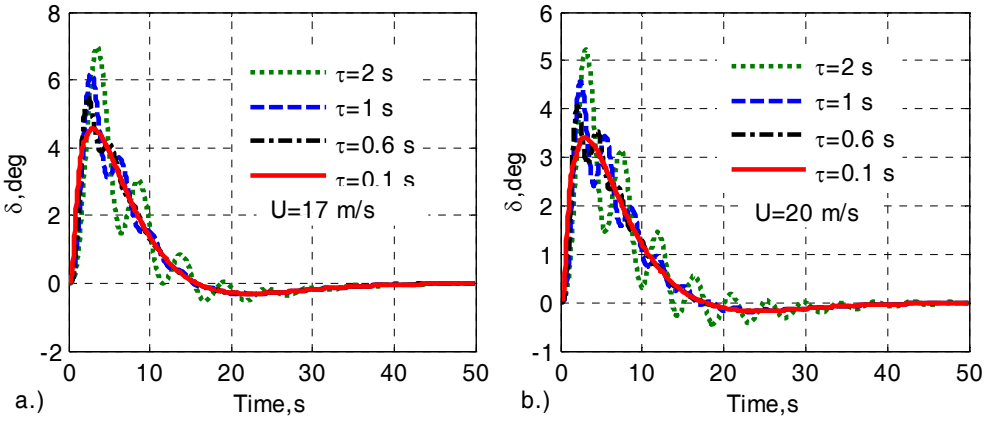

Fig. 7.11 Rudder angles for a range of different τ values (Ψ_d = 50 deg)

For the better representation of the results, only the first-half of the time histories (50 s) are included. Fig. 7.10 and Fig. 7.11 show that the oscillations in the rudder angle disappear if the time constant τ value becomes small enough. The explanation is obvious in that the channel from δ_c and δ will involve a smaller phase lag as the τ value is made smaller.

7.4 Design of a FBC for the helicopter model

In this subsection the linear Lynx-like helicopter model, introduced in Chapter 3, is considered as a benchmark for the design of a FBC. The state variables and input variables are the same as those in the previous description of the model. The main difference is that in this case the atmospheric turbulence is modelled as gust velocity components to perturb the velocity states *u*, *v*, and *w* by $\boldsymbol{d} = [\boldsymbol{d}_1 \quad \boldsymbol{d}_2 \quad \boldsymbol{d}_3]$ as in the following equations:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}\begin{bmatrix} 0 \\ \boldsymbol{d} \end{bmatrix} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{7.9}$$

In addition, the output channels are selected as shown in Table 7.1:

Table 7.1 Output variables for the Westland Lynx linearised helicopter model

Output Variables	Description	Unit
$\dot{H}$	Heave velocity	$ft \cdot s^{-1}$
θ	Pitch attitude	*rad*
Φ	Roll attitude	*rad*
$\dot{\Psi}$	Heading rate	$rad \cdot s^{-1}$
p	Roll rate	$rad \cdot s^{-1}$
q	Pitch rate	$ft \cdot s^{-1}$

The design scheme is similar to Fig. 7.1 but a MIMO situation is considered. The latest work relating to this control algorithm in the helicopter field of application can be found in (Luo *et al*., 2003; Postlethwaite *et al*., 2005). The following are the chosen values for the weighting functions $\boldsymbol{W}_1(s)$, $\boldsymbol{W}_2(s)$, and $\boldsymbol{W}_3(s)$:

$$\boldsymbol{W}_1(s) = diag\{0.2\cdot\frac{s+8}{s+0.01}, 0.5\cdot\frac{s+4}{s+0.005}, 0.5\cdot\frac{s+4}{s+0.005}, 0.5\cdot\frac{s+8}{s+0.001}, 0.5\cdot\frac{s}{s+0.001}, 0.5\cdot\frac{s}{s+0.001}\} \tag{7.10}$$

$$\boldsymbol{W}_2(s) = diag\{0.2\cdot\frac{s+0.0005}{s+10}, 0.5\cdot\frac{s+0.0005}{s+10}, 0.5\cdot\frac{s+0.0005}{s+10}, 0.2\cdot\frac{s+0.0001}{s+10}\} \tag{7.11}$$

$$\boldsymbol{W}_3(s) = diag\{1, \quad 0.1, \quad 0.1, \quad 1, \quad 1, \quad 1\} \tag{7.12}$$

The bandwidths of the weighting functions $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are selected to be around 11 rad/s by referring to the bandwidth of the disturbance model $\boldsymbol{G}_d$ (10 rad/s), and the unmodelled rotor dynamics beyond 10 rad/s. The reasons for selecting $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ have been given in the above subsection. The reasons for the choice of $\boldsymbol{W}_3$ are based on the fact that only the channels $\dot{H}$, p, q, and $\dot{\Psi}$ are selected to be outputs. The selected channels are weighted by setting their values to be one and the others are forced to be relatively unimportant by setting their values to 0.1. With these weighting-function values, the final calculated cost-function value (γ) is 1.75. $\boldsymbol{W}_4$ is the unit matrix with compatible dimensions. Fig. 7.12 and Fig. 7.13 show the quality of the performance of the designed H_∞ controller for the weighting functions selected.

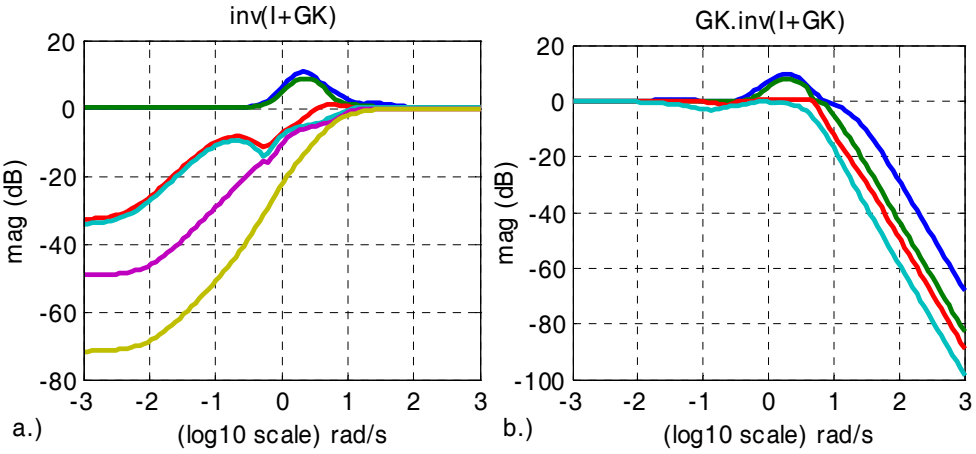

Fig. 7.12 The plot of $(\mathbf{I}+\boldsymbol{GK})^{-1}$ (a) and $\boldsymbol{GK}(\mathbf{I}+\boldsymbol{GK})^{-1}$ (b)

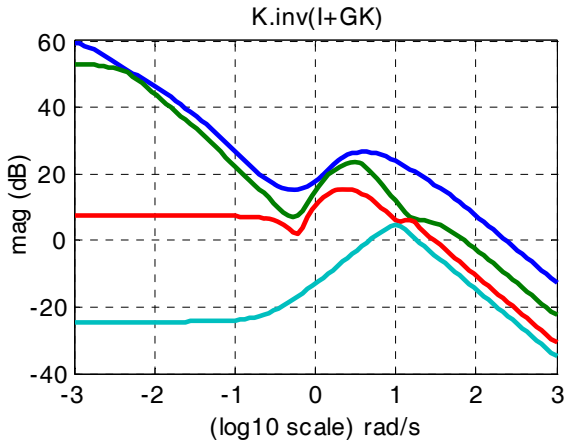

Fig. 7.13 The plot of $K(I+GK)^{-1}$

The bandwidths achieved in the results shown in Figs. 7.12 and 7.13 are both close to the expected design specification of 10 rad/s. The two singular values (p and q channels) in Fig. 7.12a which present the constant values (0 dB) in the low-frequency range results from the coupling between the aircraft attitudes and angular rates (Yue & Postlethwaite, 1990). The other four channels show small amplitudes due to being directly controlled. In addition, the sharp decrease in Fig. 7.12b in magnitude in the high-frequency range means that the good robust performance has been achieved in the controller against uncertainties resulting from unmodelled rotor degrees of freedom.

In the current investigations, two groups of manoeuvres are considered as the ideal tracking trajectories. The first group is taken from the standard heave axis response (Walker & Postlethwaite, 1996) and redefined based on the latest version of ADS-33E-PRF (Anon, 2000). The details relating to the definition can be found in Chapter 3. The selection of this group of manoeuvres is intended to provide a check on the validity of the proposed method in which the FBC is designed using the H_∞ algorithm. Compared with the first group, the second group is more demanding and has been chosen to facilitate the investigation of the tracking performance of the FBC system for severe manoeuvres. Therefore, this group is not based on standard ADS-33E height-response manoeuvres and may therefore lack the practical significance, in terms of handling qualities and agility measures, of the manoeuvres within the first group. The step response of a standard second-order transfer

function, as shown in Eq. (3.5), is still used to define the heading rate. The other three output channels are set to be the step response of the transfer function of Eq. (7.13):

$$g(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{7.13}$$

where $\xi = 0$ and $\omega_n = 3,\ 2,\ 1.5$ rad/s for p, q, and $\dot{\Psi}$, respectively. The consequent results from these applications are shown in Fig. 7.14 and Fig. 7.15.

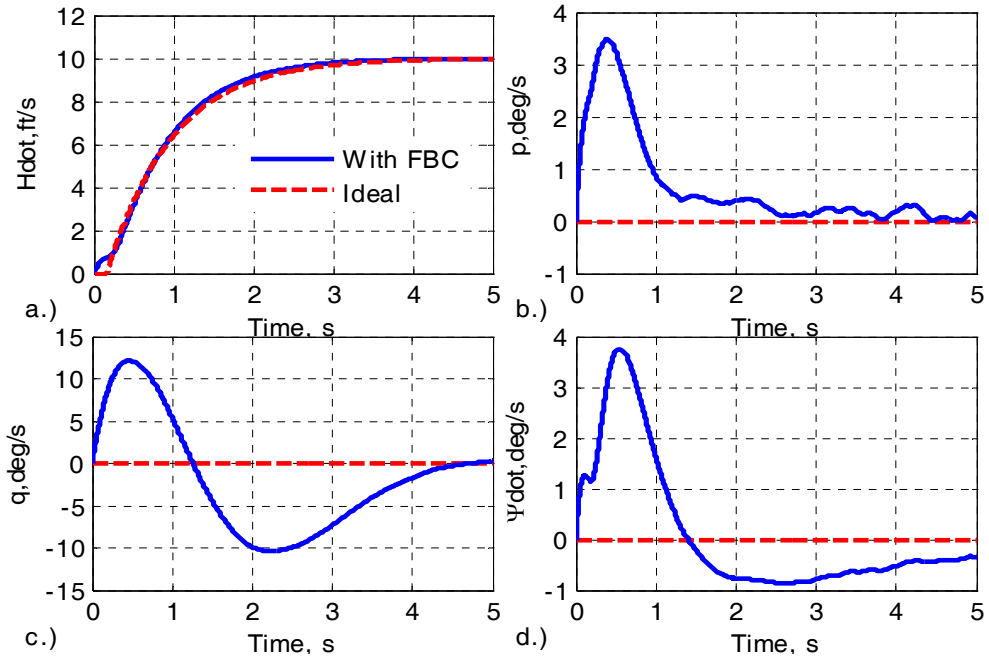

Fig. 7.14 Output tracking of ADS-33E height-response manoeuvre with measurement noise and disturbance effects

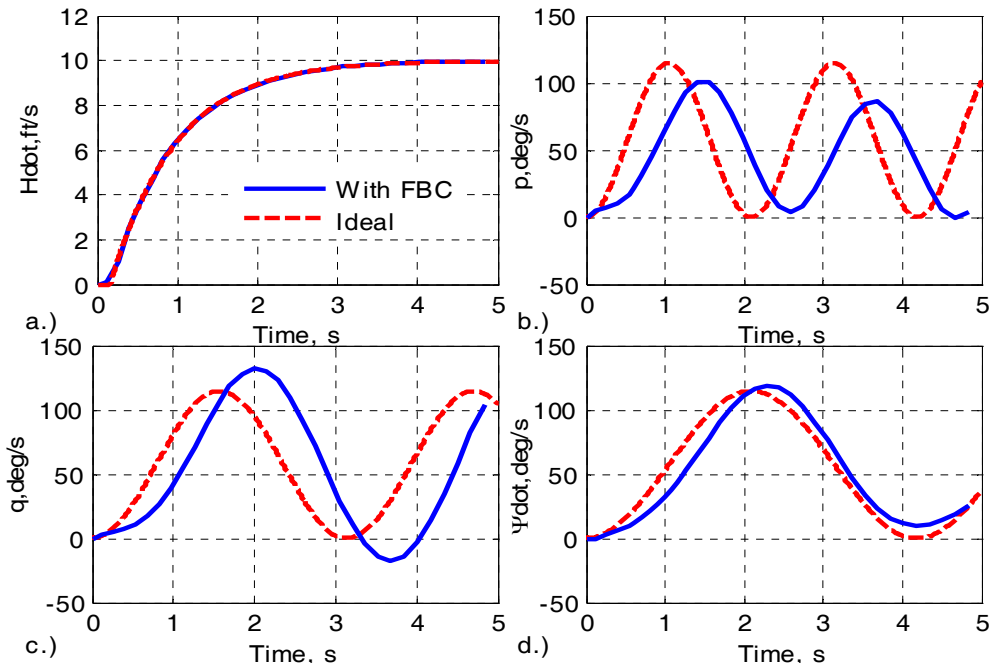

Fig. 7.15 Output tracking of typical demanding manoeuvre with measurement noise and disturbance effects

Fig. 7.14 shows that the controlled system provides good tracking performance for the standard manoeuvre considered, especially for the heading rate in which nearly perfect tracking is achieved. The tracking performance in the other three channels is slightly poorer since there are large transients during the first second or so of the records but the responses later converge to the ideal values. These results prove the effectiveness of the designed FBC for the standard ADS-33E height-response trajectories. However, for the severe trajectories, the controlled system only achieves good tracking for the heave velocity (Fig. 7.15a), and fails to reach good consistency with the ideal trajectories for the other three channels, especially for the roll rate (Fig. 7.15b) and pitch rate (Fig. 7.15c) channels in which the divergence is very evident. Moreover, this divergence is still present in the long-term simulation, from which the results are ignored here due to their similarity to the results in Fig. 7.15. This suggests there is a need for more effort to improve the tracking performance and this will be further investigated in Chapter 7 by introducing the FFC designed from the inverse simulation procedure.

7.5 Design of a FBC for the Container ship model

Much research effort has been devoted to the design of controllers which can achieve better performance in surface ships, other types of marine vessels and other forms of vehicle. Examples include the investigation of a general control system for trajectory tracking of an under-actuated ship (Do & Pan, 2006), an autopilot system for the safe navigation of a surface vehicle (Alfaro-Cid *et al*., 2005), and systems for ship steering in waves (Fang & Luo, 2005) etc. In addition, knowledge of nonlinear control theory also has been applied to solving problems of steering of vehicles (Hu *et al*., 2003; Tzeng, Goodwin & Crisafulli 1999).

This section tackles the problem of combined ship roll stabilisation and tracking, by means of the LQ control approach and the mixed-sensitivity H_∞ control methodology. The implementation of a LQ controller outlined in this section is intended to provide a reference for comparison with the results from the mixed-sensitivity H_∞ method. This section first introduces the design of the controllers for a rudder-roll stabilisation system (RRS) using the Son and Nomoto Container ship model in the linearised form. The nonlinear counterpart of this model has been implemented in Chapter 6 to validate the constrained NM-based method for inverse simulation. These designed controllers are subsequently tested using the full nonlinear Container ship model. This allows comparisons to be made and provides a basis for some concluding remarks.

7.5.1 Design of controllers and simulation with the linear model

A rudder-roll subsystem involving the linearised form of the Son and Nomoto Container ship model (Escande, 1997; Fossen *et al*., 2005), provides a benchmark for designing the controllers. This linear RRS involves the sway, roll, and yaw variables and associated cross-couplings and its structure can be described by the following equations:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{d} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{aligned} \tag{7.14}$$

where $\boldsymbol{x} \in \mathbb{R}^5$ is the state-variable vector, which consist of the sway velocity (v), the roll velocity (p), the yaw velocity (r), the roll angle (Φ), and the heading angle (Ψ). $\boldsymbol{y} \in \mathbb{R}^5$ is the vector of output variables which are the five state variables used for feedback. For the applications described in this section, all these five state variables are required to design a FBC. If they cannot be measured directly, a Kalman filter can be used to estimate the unknown values. The input $\boldsymbol{u} \in \mathbb{R}^1$ is the rudder motion and $\boldsymbol{d} \in \mathbb{R}^1$ is a random disturbance.

7.5.1.1 Design of a linear quadratic controller

Because it is a well-known approach, the details of the LQ controller are ignored here and the interested reader can refer to the book (Fossen, 1994). The main structure of this algorithm is shown in Fig. 7.16:

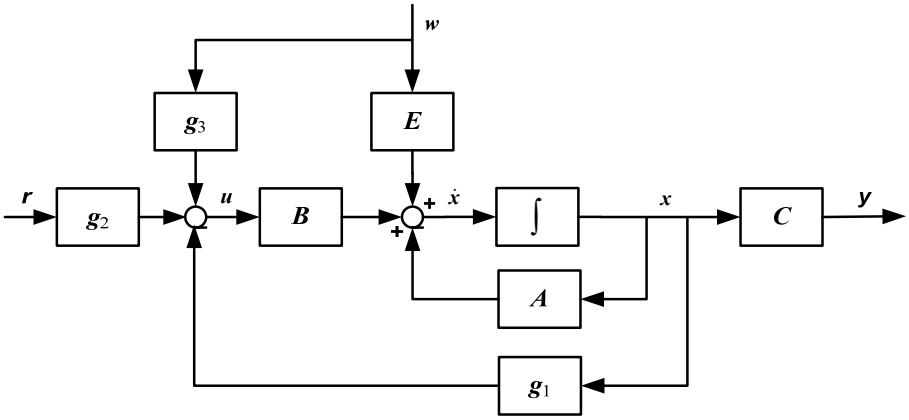

Fig. 7.16 Diagram showing the Linear Quadratic Optimal Control system

In Fig. 7.16, $\boldsymbol{r}$ is the given reference vector. During the design stage of the LQ controller, the disturbance term $\boldsymbol{d}$ in Eq. (7.14) is ignored. After calculation using standard LQ methods, the values of $\boldsymbol{r}$, $\boldsymbol{g}_1$, and $\boldsymbol{g}_2$ are as follows:

$$\begin{aligned} \boldsymbol{r} &= \begin{bmatrix} p & r & \Phi & \Psi \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 10 \end{bmatrix} \\ \boldsymbol{g}_1 &= \begin{bmatrix} 0.1631 & -16.1193 & -6.7655 & -1.1644 & -0.4472 \end{bmatrix} \\ \boldsymbol{g}_2 &= \begin{bmatrix} 0 & 0 & 0 & 0.4472 \end{bmatrix} \end{aligned} \tag{7.15}$$

To show the performance of the heading tracking system as well as the rudder-roll decoupling in the presence of measurement noise but no wave disturbances, the following two situations have been investigated. Firstly the performance has been investigated without disturbances and measurement noise, and secondly the performance has been considered in the absence of disturbances but with measurement noise. The measurement noise is a Gaussian white noise, which is added to each feedback channel at the measurement point as shown in Fig. 7.16. In order to highlight the action of the control system during the simulation process, the feedback channels of the roll rate (p) and roll angle (Φ) are turned on after 300 s and off again after 500 s. The performance against wave disturbances will be investigated in the next section.

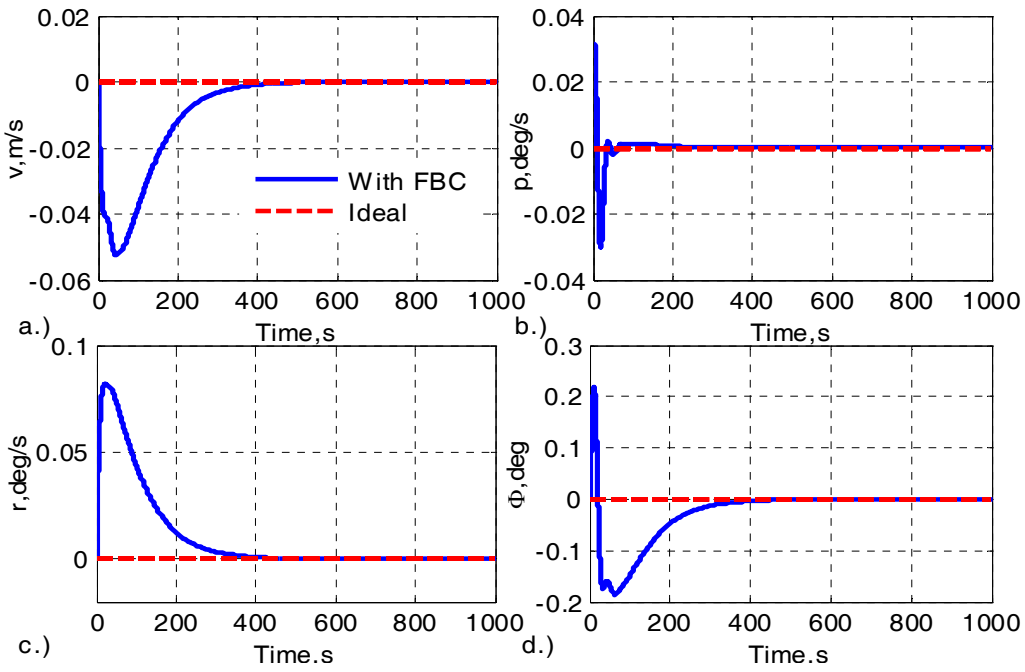

Fig. 7.17 Results from the LQ controller without disturbance and measurement noise (linear model)

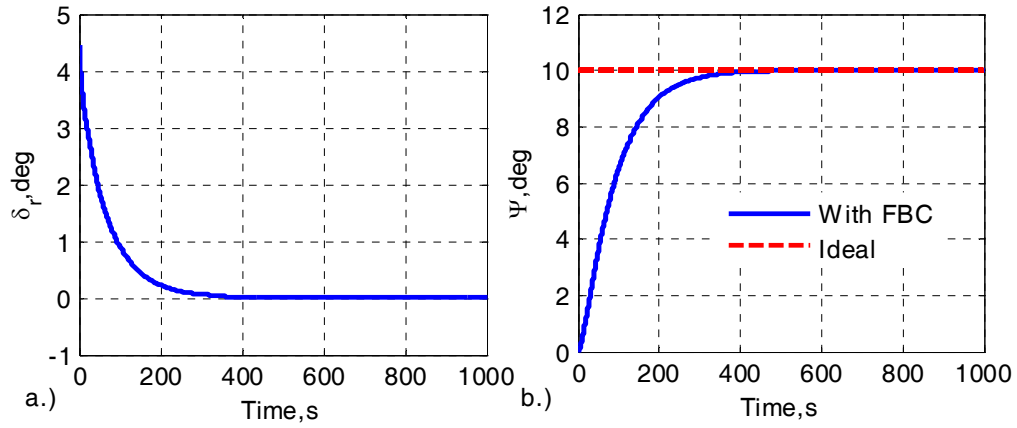

Fig. 7.18 Results from the LQ controller without disturbance and measurement noise (linear model)

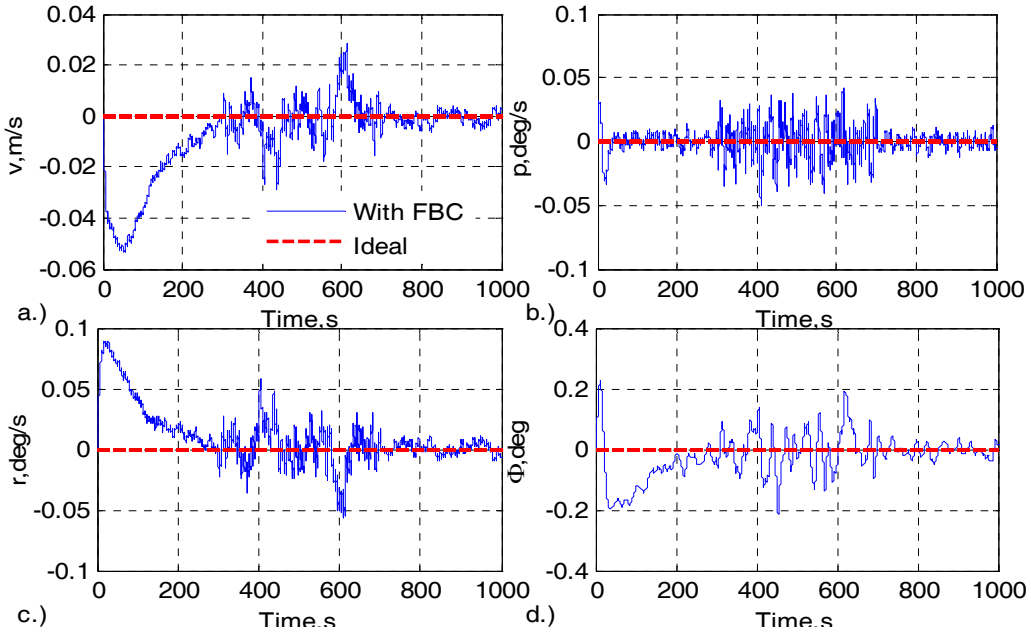

Fig. 7.19 Results from the LQ controller without disturbance but with measurement noise (linear model)

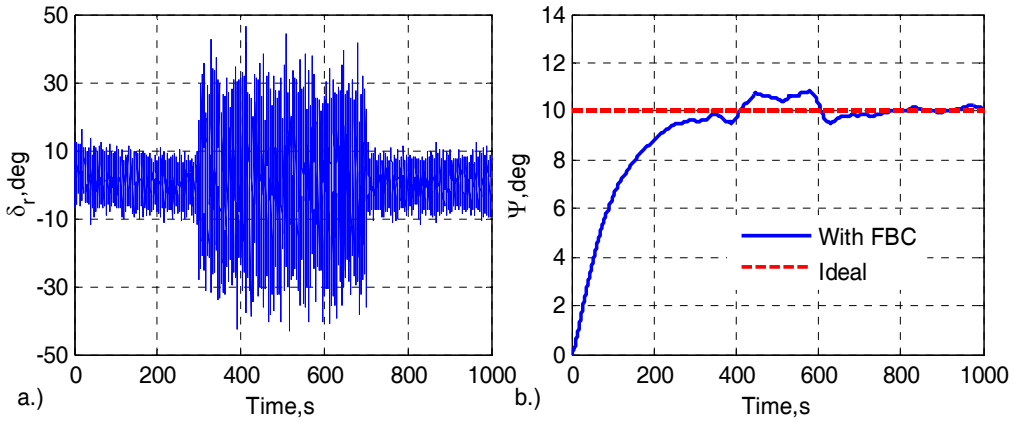

Fig. 7.20 Results from the LQ controller without disturbance but with measurement noise (linear model)

Fig. 7.19 and Fig. 7.20 show that the performance in terms of the heading tracking and the rudder-roll decoupling with measurement noise is relatively poor compared with Fig. 7.17 and Fig. 7.18 which show the case without measurement noise. Moreover, the introduced measurement noise has a significant influence on the heading tracking accuracy and the control effort required, as shown in Fig. 7.20, where the input to the model reaches a maximum magnitude of 40 deg during the period from 300 s to 500 s. This value is far larger than the saturation level of 20 deg, and is therefore unacceptable. This large control effort required is due to the fact that the effects of the measurement noise can be fed back to the input directly and in a linear proportional fashion, as shown in Fig. 7.16. In addition, the feedback of the channels p and Φ during the period from 300 s to 500 s has a significant influence on the heading tracking accuracy and the control effort for this linear case, as shown in Fig. 7.20. Therefore, this application case highlights some weak points of the approach based on the LQ controller.

7.5.1.2 Design of a mixed-sensitivity H_∞ controller

The mixed-sensitivity H_∞ control achieves robust stability and robust performance through transfer function shaping by selecting suitable weighting functions (Skogestad & Postlethwaite, 1996). The standard procedure to design an H_∞ controller introduced in Section 7.3 is applied in this section and the only difference is that the calculation of $\boldsymbol{B}_{dt}$ in

Eq. (7.4) depends on the dynamic parameters of the nonlinear Container ship model being investigated for this case. By following the similar procedure to select the weighting functions, the following are the chosen values for the reference inputs $\boldsymbol{r}$ and the weighting functions $\boldsymbol{W}_{1\text{-}4}(s)$:

$$\boldsymbol{W}_1 = diag\{\ 0.05\cdot\frac{s+0.12}{s+0.05}\quad 0.05\cdot\frac{s+0.12}{s+0.05}\quad 0.05\cdot\frac{s+0.12}{s+0.05}\quad 0.02\cdot\frac{s+0.1}{s+0.001}\quad 0.04\cdot\frac{s+0.06}{s+0.001}\ \} \tag{7.16}$$

$$\boldsymbol{W}_2 = 0.12\cdot\frac{s+0.001}{s+0.3} \tag{7.17}$$

$$\boldsymbol{W}_3 = diag\{\ 1\quad 1\quad 1\quad 1.1\quad 1\ \} \tag{7.18}$$

$$\boldsymbol{W}_4 = 0.6 \tag{7.19}$$

$$\boldsymbol{r} = [0\quad 0\quad 0\quad 0\quad 10] \tag{7.20}$$

The bandwidths of the weighting functions $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are selected to be less than 0.3 rad/s by referring to the bandwidth of the original model $\boldsymbol{G}$ (0.19 rad/s) and the bandwidth (0.2 rad/s) of the closed-loop model with the LQ controller designed in the previous section, as well as the rudder control and rate constraints. The inclusion of the integral action in $\boldsymbol{W}_1$ is used to compensate for wave drift (2^{nd}-order-wave-induced motion) and low-frequency ocean current disturbances. The cost-function value (γ) of the H_∞ norm of Eq. (7.3) is 2.2.

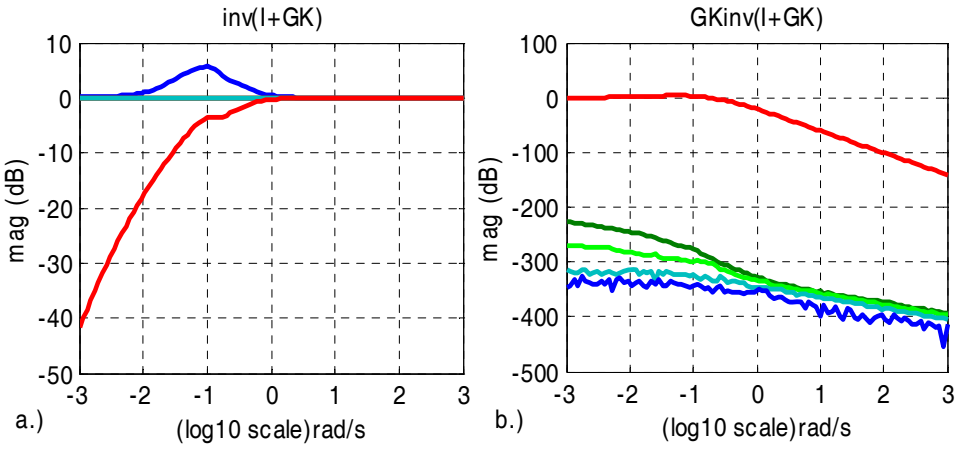

Fig. 7.21 The plots of $(I+GK)^{-1}$ (a) and $GK(I+GK)^{-1}$ (b)

Fig. 7.21a shows a plot of the singular values of the sensitivity function (***S***), which determines the performance of disturbance rejection in the low-frequency range. Because of the five-by-five size of the matrix ***S*** and the fact that there is only one-input channel, four singular values (corresponding to v, p, r and Φ) will always have a value of one (corresponding to 0 dB on a log scale) across all frequencies and the singular value corresponding to the heading (Ψ) decreases rapidly in the frequency range of 0 to 1 rad/s (the red line), which is consistent with the bandwidth of the disturbance model. The small bump existing in one of the singular-value plots is associated with a NMP phenomenon. Fig. 7.21b shows the singular-value plot of the complementary sensitivity function (***T***), which is useful for determining the tracking performance. In that figure, the singular value of the heading channel (the red line) has a value of one within the low-frequency band, which suggests that good tracking performance may be achieved for this channel. However, for the other four channels, the singular values across the low frequencies are much smaller than one. This complies with the difficulty encountered in simulation studies for the Container ship system to obtain good tracking for all channels simultaneously. The results from the simulation are shown in Fig. 7.22 and Fig. 7.23.

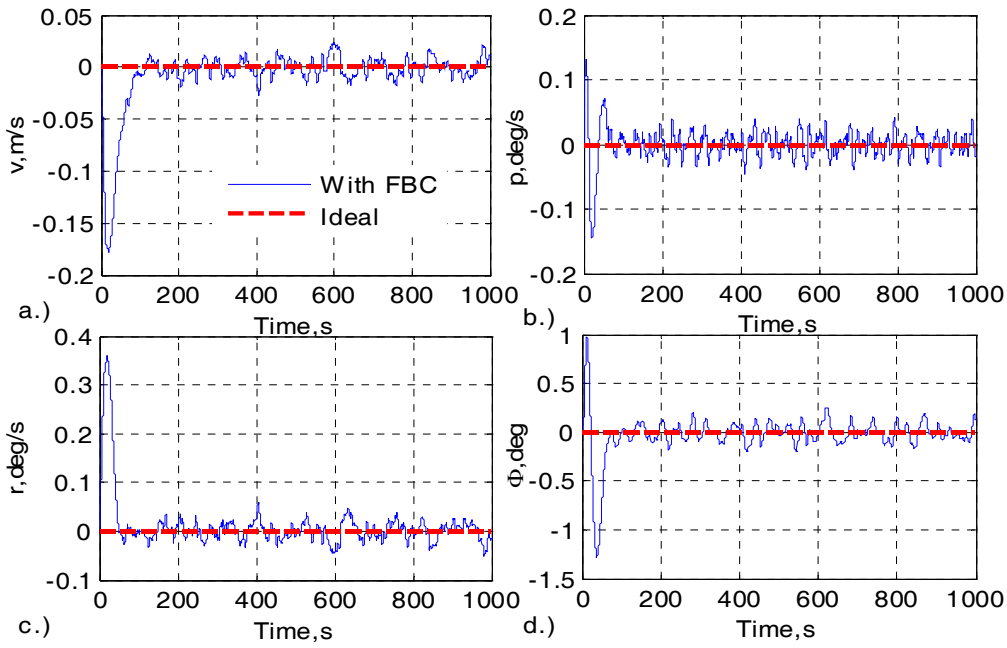

Fig. 7.22 Results from the *K*/*KS* controller with disturbance and measurement noise (linear model)

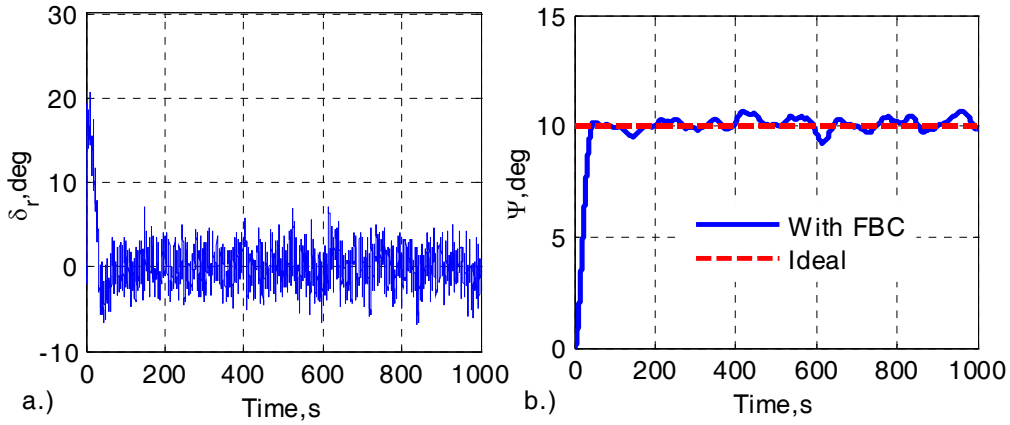

Fig. 7.23 Results from the *K*/*KS* controller with disturbance and measurement noise (linear model)

From Fig. 7.22 and Fig. 7.23, it can be seen that the *K/KS* controller achieves good tracking performance and disturbance rejection in the presence of disturbances and measurement noise. In addition, the feedback of the channels p and Φ during the period from 300 s to 500 s has almost no influence on the heading tracking and the control effort for this linear case, as shown in Fig. 7.23. However, the control is affected by high-frequency oscillations but is less noisy compared with the results from the LQ controller. One thing that the reader has to keep in mind is that the controller designed above may be not optimal although the γ value is quite small. The reason for selecting the terms $\boldsymbol{W}_{1\text{-}4}$ listed above is that they can provide good performance both for the linear system and the nonlinear system which will be discussed in later sections.

7.5.2 Simulation using the nonlinear model

In this section, the controllers designed in the above sections are applied to the full nonlinear Son and Nomoto Container ship model with wave-induced roll-moment disturbances (Fossen *et al.*, 2005). This model has twelve state variables and two inputs. The input constraints, as well as other necessary information, can be found in the given reference.

Four forward speed (U) situations: 5 m/s, 7.3 m/s, 10 m/s, and 13 m/s, are simulated to validate the robust performance and stability of the designed controllers, which are designed based on the model linearised around the forward speed 7.3 m/s. For these four situations, the tests both with or without measurement noise are also carried out to show the effectiveness of the *K*/*KS* approach against measurement noise, though a lot of approaches (such as the Kalman filter) are available for computing noise-free estimates of the states. Results from the other three forward speeds are not presented here due to the fact that they are very similar to those from the forward speed 7.3 m/s.

The Matlab function *wgn* is used to generate the Gaussian white measurement noise with output power 1 dBW and load impedance 0.005 Ohms. In addition, the feedback channels v, r, and Ψ and the roll-moment channels p and Φ are filtered by a low-pass filter and a high-pass filter, as shown respectively in Eq. (7.21) and Eq. (7.22), to eliminate the effect of the high-frequency motion to counteract the 1st-order wave disturbances and also to avoid causing additional wear and tear on the thruster actuators (Fossen, 1994):

$$f_L(s) = \frac{1}{s + 0.1} \tag{7.21}$$

$$f_h(s) = \frac{s}{s + 0.05} \tag{7.22}$$

However, in the present work on H_∞ control system synthesis to demonstrate the improved performance of the H_∞ controller, the above wave filtering is not included in the feedback channels. The controllers are synthesised in such a way that they provide the necessary filtering action. For all simulations, the feedback of the channels p and Φ are switched on over the period from 300 s to 500 s and off at all other times.

7.5.2.1 Application involving the LQ controller

a.) Simulated results without measurement noise

The LQ controller designed in Section 7.5.1.1 is tested on the nonlinear Container ship model. The results are shown in Figures 7.24 – 7.25 as follows:

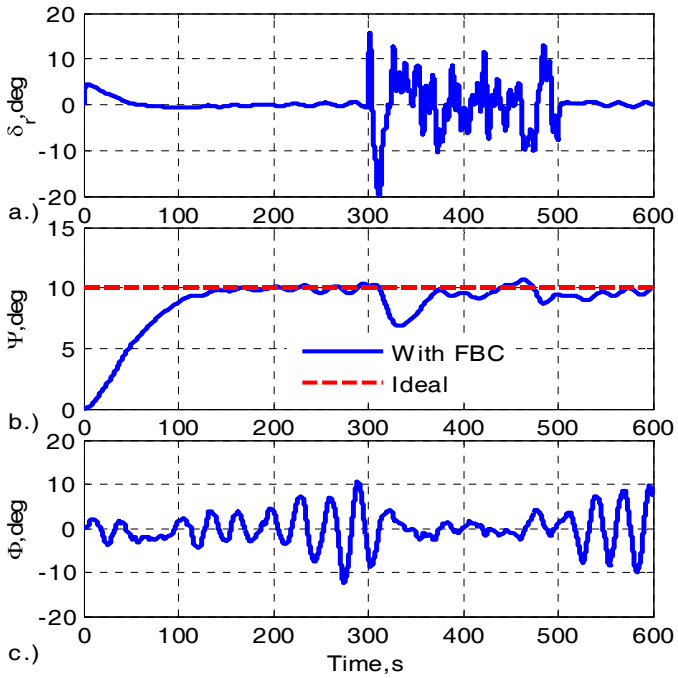

Fig. 7.24 Results from the LQ controller with disturbance but without measurement noise (nonlinear model, U_0 = 7.3 m/s)

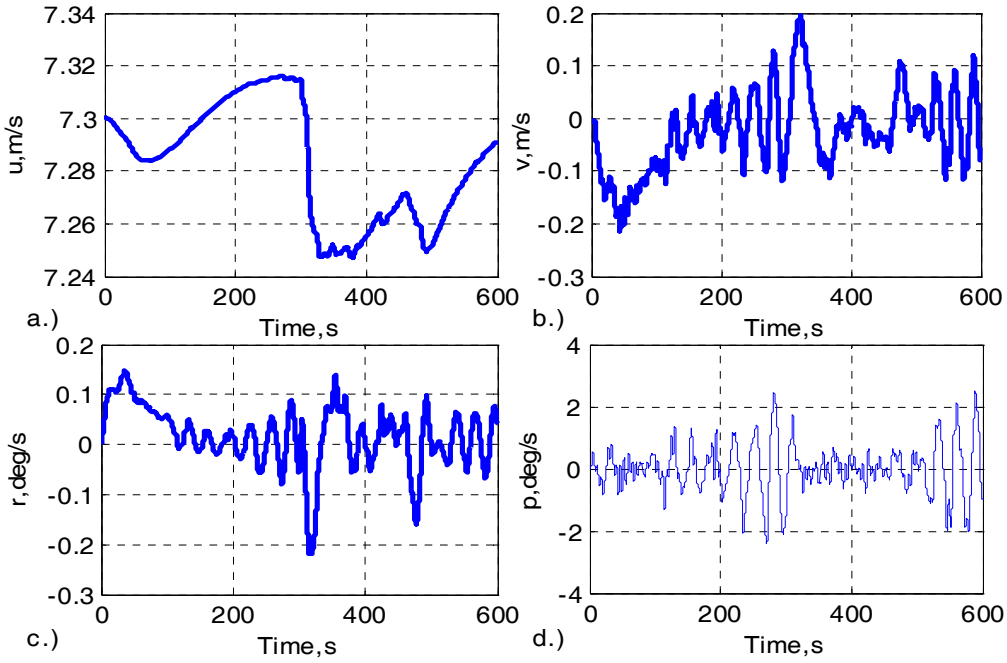

Fig. 7.25 Results from the LQ controller with disturbance but without measurement noise (nonlinear model, U_0 = 7.3 m/s)

The root-mean-square (RMS) measure is used to quantitatively establish the effectiveness of the rudder-roll reduction for the roll angle (Φ) and the roll rate (p) channels. The calculated RMS values are:

$$\Phi_w = 3.76 \text{ deg} \quad p_w = 0.80 \text{ deg/s} \quad \Phi_b = 2.27 \text{ deg} \quad p_b = 0.42 \text{ deg/s} \tag{7.23}$$

where the subscript w represents the whole simulation period and b the period when the feedback channels Φ and p are switched on. These values show that the roll angle (Φ) has been reduced by 1.49 deg and also slightly for the roll rate (p), as shown in Fig. 7.24c and Fig. 7.25d. However, Fig. 7.24 also shows the deterioration in heading tracking performance (b) as well as the significantly increased magnitude of the rudder angle (a). This rudder angle is generated inside the nonlinear model and has nearly reached the saturation level (20 deg) at some points. In addition to this, the input $\boldsymbol{u}$ to the nonlinear model is quite large and also is shows high-frequency oscillations, as shown in Fig. 7.26:

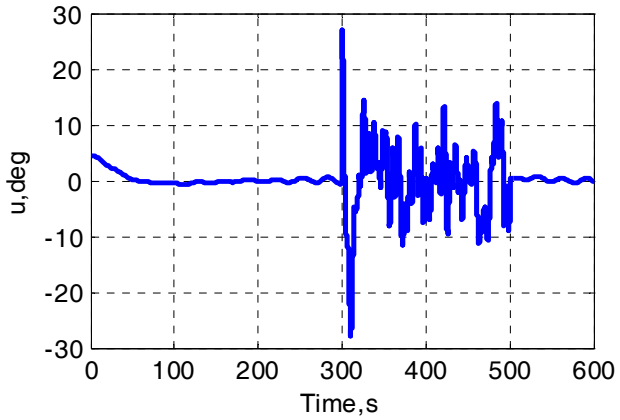

Fig. 7.26 Inputs to the nonlinear model with disturbance but without measurement noise (LQ controller, U_0 = 7.3 m/s)

The first two spikes in Fig. 7.26 have exceeded the saturation level (20 deg). Furthermore, the high-frequency oscillations existing in the input ($\boldsymbol{u}$) to the nonlinear model have resulted in oscillating movements in the rudder deflection δ_r, as shown in Fig. 7.24a. This is believed to have some negative effects such as causing wear and tear. All this means that the rudder-roll reduction is achieved by the LQ controller at the cost of introducing extra control effort with additional high-frequency oscillations as well as the poorer tracking performance. The fuel consumption therefore may be increased.

b.) Simulated results with measurement noise

The simulation with the measurement noise in the LQ control synthesis aims to show the poor performance of the LQ controller in the presence of measurement noise. The obtained results are shown as follows in Figs. 7.27 – 7.29:

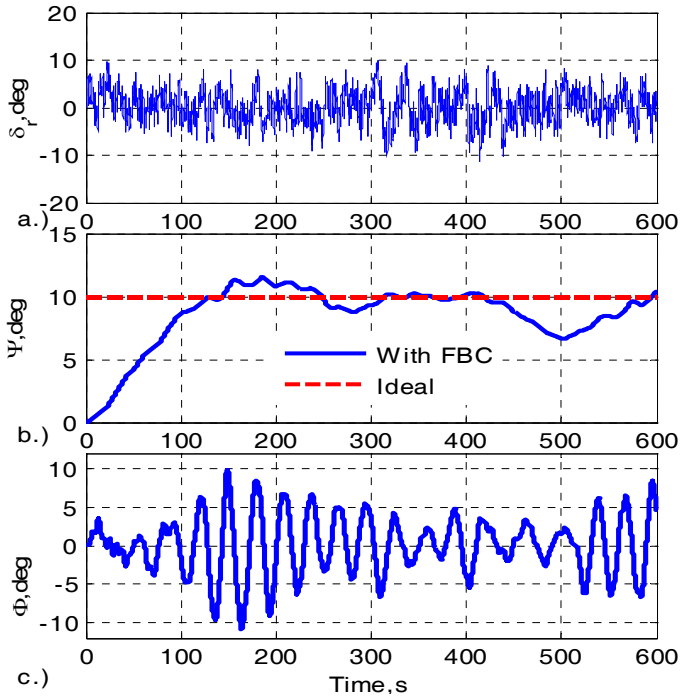

Fig. 7.27 Results from the LQ controller with disturbance and measurement noise (nonlinear model, U_0 = 7.3 m/s)

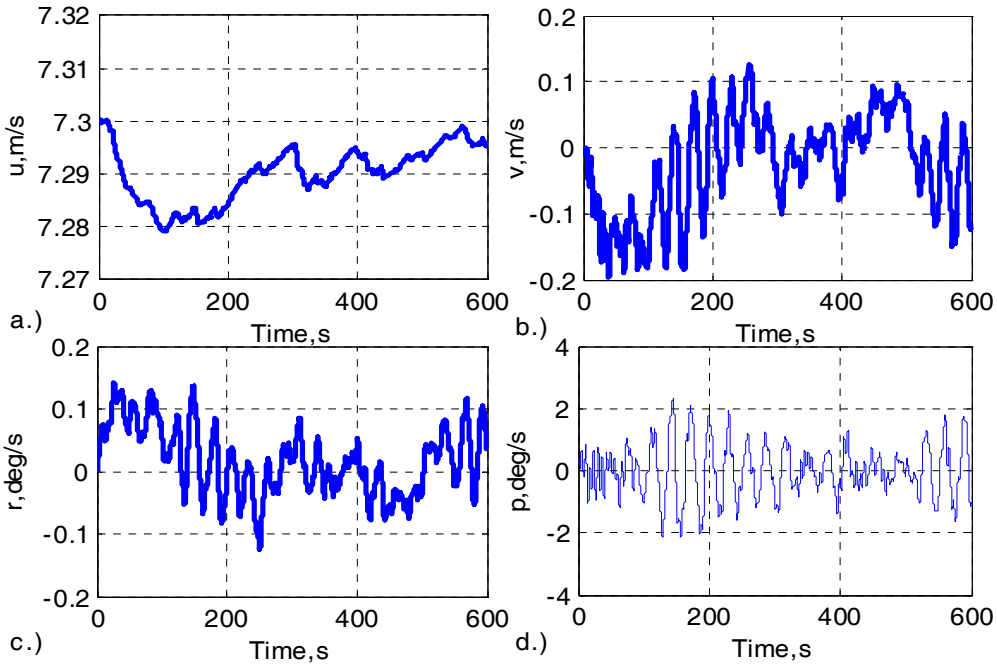

Fig. 7.28 Results from the LQ controller with disturbance and measurement noise (nonlinear model, U_0 = 7.3 m/s)

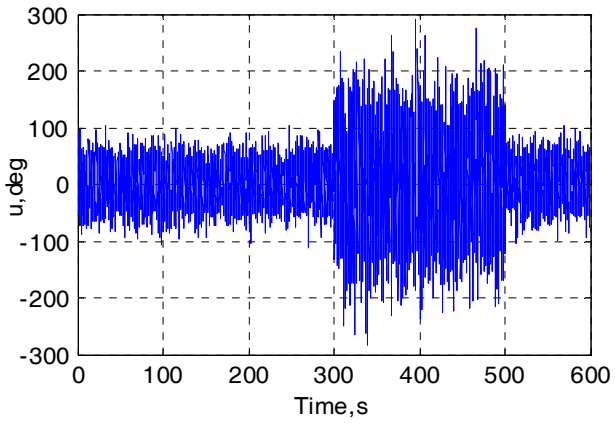

Fig. 7.29 Inputs to the nonlinear model with disturbance and measurement noise (LQ controller, U_0 = 7.3 m/s)

In this case, the calculated RMS values for channels p and Φ are shown in Eq. (7.24):

$$\Phi_w = 3.82 \text{ deg} \quad p_w = 0.84 \text{ deg/s} \quad \Phi_b = 2.43 \text{ deg} \quad p_b = 0.54 \text{ deg/s} \tag{7.24}$$

Although the rudder-roll control system takes effect, the rudder angle generated within the nonlinear model is affected by high-frequency oscillations, as shown in Fig.7.27a. These

frequencies are well above the frequency range of the wave disturbances. The values in Eq. 7.24 show that the roll angle (Φ) has been reduced by 1.39 deg and is also slightly reduced for the roll rate (p), as shown in Fig. 7.27c and Fig. 7.28d. Moreover, the magnitudes of the input to the model shown in Fig. 7.29 are unacceptably large. The magnitude values mean the rudder actually always works at the saturation level and this has been proved by the simulation process.

The reason of the poor performance of the LQ control system in suppressing the measurement noise is obvious due to the fact that the channel between the input $\boldsymbol{u}$ and the measuring point, shown in Fig. 7.1, only involves a linear multiplier factor $\boldsymbol{g}_2$. Therefore, the measurement noise will be directly reflected in the input to the nonlinear model.

7.5.2.2 Application involving the *K*/*KS* Controller

The mixed-sensitivity controller based on the H_∞ algorithm has been tested on the nonlinear Container ship model with and without measurement noise. Results are presented in this section. Again, the feedback of the channels p and Φ are switched on for the period of time from 300 s to 500 s and are switched off at all other times.

a.) Simulated results without measurement noise

The H_∞ controller designed in Section 7.5.1.2 is tested on the nonlinear Container ship model. The results are shown in Figs. 7.30 – 7.31 as follows:

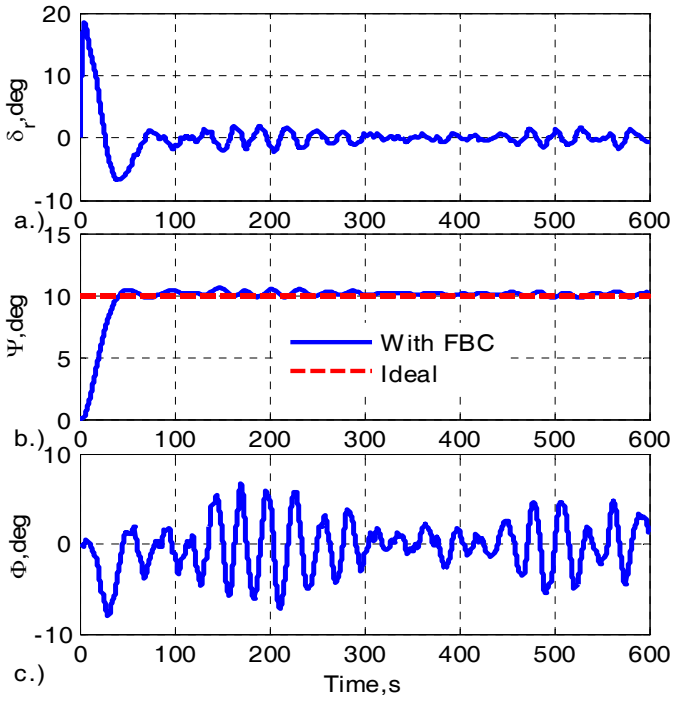

Fig. 7.30 Results from the *K*/*KS* controller with disturbance but without measurement noise (nonlinear model, U_0 = 7.3 m/s)

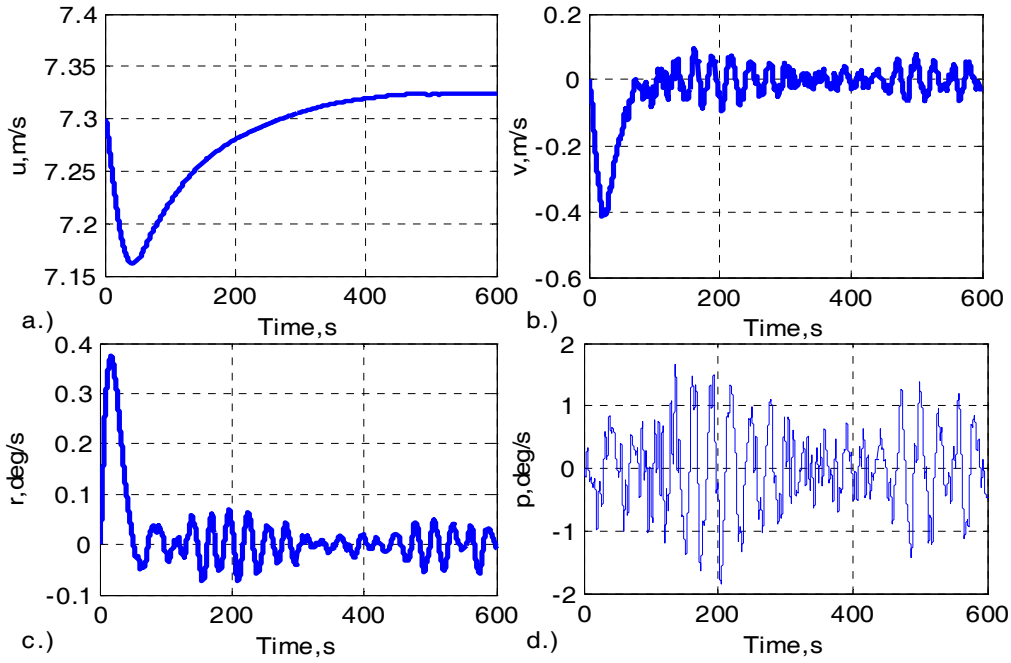

Fig. 7.31 Results from the *K*/*KS* controller with disturbance but without measurement noise (nonlinear model, U_0 = 7.3 m/s)

The calculated RMS values for the channels p and Φ are shown in Eq. (7.25):

$$\Phi_w = 2.86 \text{ deg} \quad p_w = 0.63 \text{ deg/s} \quad \Phi_b = 1.88 \text{ deg} \quad p_b = 0.46 \text{ deg/s} \qquad (7.25)$$

These values show that the roll angle (Φ) has been reduced by 0.98 deg as well as the roll rate (p), as presented in Fig. 7.30c and Fig. 7.31d. Compared with the results obtained by the LQ method, the *K*/*KS* controller achieves nearly perfect heading tracking although the rudder angle is quite large in the first few seconds. However, the rudder is still operating within the saturation limits. In addition, the *K*/*KS* approach also shows that the roll angle (Φ) is reduced during the period when the feedback channels p and Φ are switched on. Moreover, the performance in terms of heading and rudder deflection remains good, unlike the performance obtained with the LQ approach. No graphical results have been included showing the input to the nonlinear model because it is very similar to the result shown in Fig. 7.30a. All these results prove the effectiveness of the *K*/*KS* H_∞ approach for this application. Again, better controllers might well be found by further adjustment of the weighting functions at the design stage.

b.) Simulated results with measurement noise

This section describes investigation of the insensitivity of the *K*/*KS* controller to measurement noise. It should be noted that this is achieved without introducing additional control complexity such as arises when filters are included in the feedback channels. The results are shown as follows in Figs. 7.32 – 7.33:

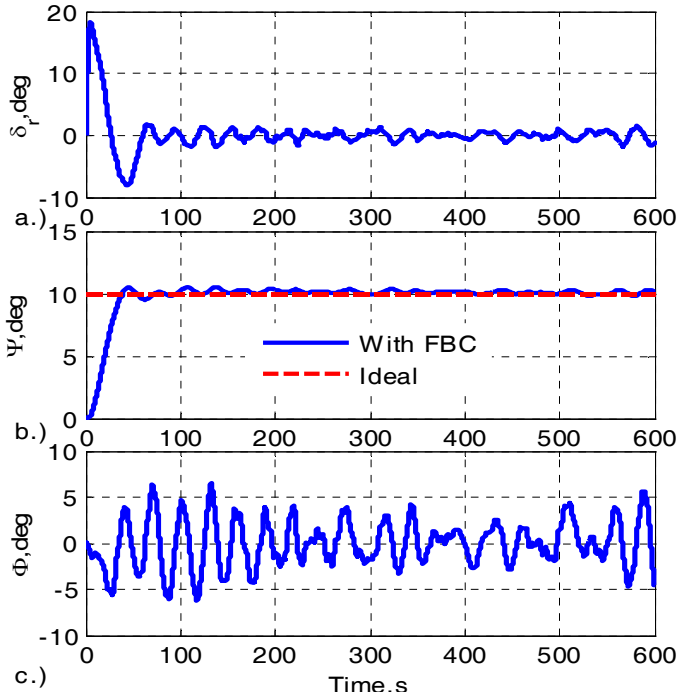

Fig. 7.32 Results from the *K/KS* controller with disturbance and measurement noise (nonlinear model, U_0 = 7.3 m/s)

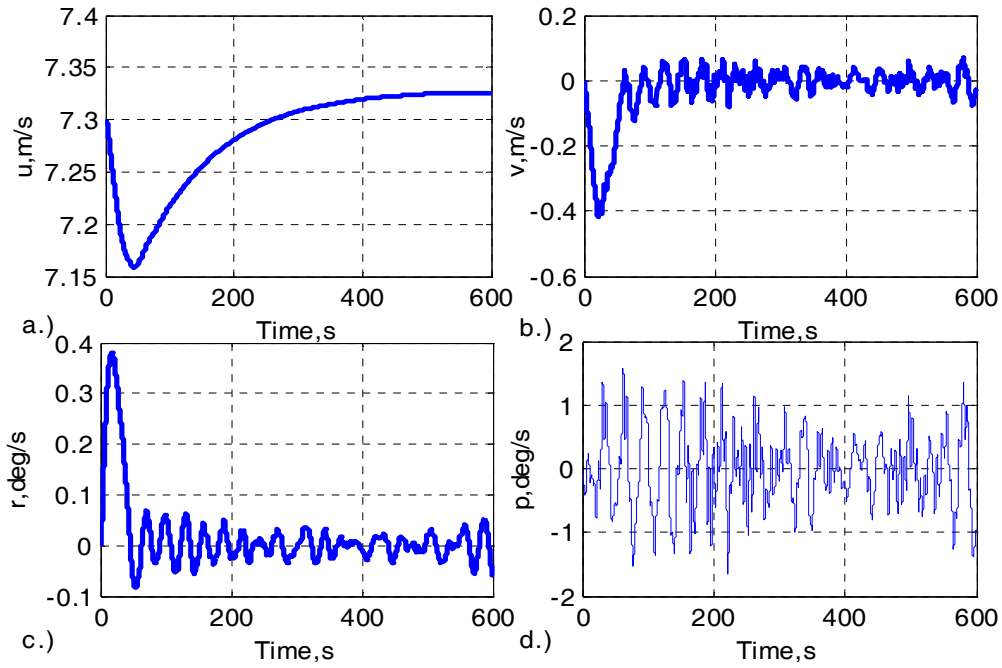

Fig. 7.33 Results from the *K/KS* controller with disturbance and measurement noise (nonlinear model, U_0 = 7.3 m/s)

The calculated RMS values for the channels p and Φ at this case are shown in Eq. (7.36):

$$\Phi_w = 2.55 \text{ deg} \quad p_w = 0.59 \text{ deg/s} \quad \Phi_b = 1.66 \text{ deg} \quad p_b = 0.41 \text{ deg/s} \tag{7.36}$$

From the RMS values and the figures shown above, the *K/KS* controller shows good performance in the presence of measurement noise. Neither the magnitude of the input to the nonlinear model is too large, nor is it affected by the high-frequency oscillations found in the LQ controller case, shown in Fig. 7.32. The input to the nonlinear model is not presented here in graphical form because of its very close similarity to the result shown in Fig. 7.32a. Furthermore, this case also achieves good heading tracking performance without being affected by the disturbances, measurement noise, and the excitation of the rudder-roll subsystem.

In fact, the good properties of the *K/KS* controller in terms of noise attenuation can be analysed by referring to Fig. 7.21. The maximum singular value of the complementary function $\bar{\sigma}(\boldsymbol{T})$ shown in Fig. 7.21b decreases rapidly for frequencies greater than 0.12 rad/s. As a result, unlike the LQ method, the extra complexity to deal with the measurement noise is not necessary in this case.

7.6 Summary

The feedback control systems designed from the mixed-sensitivity H_∞ optimisation method show perfect tracking performance as well as good robust stability against external disturbances and insensitivity to measurement noise, for the three different models considered – the Norbbin ship model, the linear Lynx-like helicopter model, and the nonlinear Container ship model. For the case of the Norbbin ship model, the results show robustness to changes of the forward speed. Oscillations in the rudder angle have been found to be related to the time constant value (τ) in the representation of the steering machine and disappear if τ is made small enough. For the Lynx-like helicopter model, the designed FBC shows good tracking performance for the standard manoeuvres based on ADS-33E recommendations but fails in the case of more severe (but more artificial) manoeuvres.

When the Son and Nomoto Container ship model is considered, the two approaches (the LQ method and the mixed-sensitivity H_∞ optimisation method) have been successfully developed. The results from the simulation of the linear and nonlinear models have validated the effectiveness of these two controllers. The implementation of the mature LQ method is the most straightforward because no weighting-function selection is involved. Therefore, the LQ method avoids trial and error methods. The mixed-sensitivity H_∞ optimisation method provides more freedom to improve the robust stability and robust performance. However, this is achieved only at the cost of added complexity in the design process, especially in terms of the choice of factors such as the weighting functions used against the disturbance and measurements noise. The difficulty in implementation of this method arises because of the fact that some performance criteria conflict with each other. An obvious example of this is the roll moment reduction and the heading tracking accuracy.

For the LQ method, the performance specified in the mathematical optimization process to obtain the controller, is well embodied in the final results in the simulation of the nonlinear model. One example of this is the reduction achieved in roll. However, this linking of the optimisation process and the final performance is not so apparent for the H_∞ algorithm based methods where it has been found that the gamma value has mathematical meaning but little significance in terms of the physical interpretation of performance improvement. In addition, the performance of the controller based on the linear model will deteriorate when that controller is applied on the nonlinear model.

The both types of controller considered show robustness to changes of forward speed for the Son and Nomoto Container ship model. This has been demonstrated in the simulation results for the four speed situations: 5 m/s, 7.3 m/s, 10 m/s, and 13 m/s. In addition to this performance robustness, the *K*/*KS* results show good performance in the presence of measurement noise. Therefore no extra design efforts are required to deal with measurement noise in this approach. Furthermore, the design achieves good rudder-roll reduction with no large variation of the magnitude of the rudder input while maintaining good tracking performance. In contrast, the LQ controller fails to achieve such a good performance overall for the case of the nonlinear ship model.

Chapter 8

Feedforward Controller Design

Contents

The primary objective of this chapter is to investigate the use of inverse simulation to develop robust feedforward tracking controllers for the traditional 2DOF output-tracking control system structure, thus avoiding the involvement of the more complicated and tedious techniques of model inversion.

8.1 Introduction

This chapter focuses on a detailed description of the use of inverse simulation techniques that have been extensively investigated in the aircraft field over the past decade (as has been introduced and developed in the previous chapters) to replace model inversion in the traditional 2DOF control structure. The previous investigations have found that, provided a suitable value of discretized time interval is used, inverse simulation is preferred to model inversion for MP systems. Moreover, unlike most currently available approaches for model inversion, inverse simulation provides an alternative more feasible and causal way to determine the required inputs to follow a predefined trajectory for a NMP system,

depending upon zero redistribution within the process of inverse simulation, as discussed in Chapter 3.

These results, combined with the H_∞ feedback controllers designed in Chapter 7, are demonstrated by applications involving an eighth-order linear Lynx-like helicopter model and a full nonlinear Container ship model used in the context of ship steering control and roll stabilization. It is believed that the conclusions from this demonstration process may help establish the validity and effectiveness of the approach based on inverse simulation to replace model inversion for design of the FFC.

This chapter first considers the issue of whether or not to introduce the FFC for different levels of uncertainty in the plant model. Then, the FFC is added and combined with the results in Chapter 7 to implement the complete 2DOF control structure. The performance of this whole system is investigated using the Lynx-like helicopter model and the Container ship model. Finally, the main features of the results from these two applications are summarized.

8.2 Uncertainties in the 2DOF structure

In the absence of plant uncertainties such as parameter uncertainties (termed structured uncertainties) and neglected and unmodelled dynamics uncertainty, it is not a challenging design problem for the 2DOF control structure reviewed in Chapter 2 to achieve perfect output tracking. However, as mentioned above, the performance and accuracy is highly dependent on the accuracy of the modelled dynamics of the controlled system. This is due to the fact that here model-based inversion methods are applied to achieve high-precision output tracking. In addition, the FFC cannot correct tracking errors resulting from plant uncertainties. Moreover, it has been shown that larger uncertainties in the controlled model lead to degraded tracking performance with the FFC approach. This raises a question about whether the FFC structure should still be adopted when the uncertainties are large. In the following, the results from previous investigations of parameter uncertainties and

unmodelled uncertainties are summarised. All these investigations are based on the use of linear system models.

The work of Zhao and Jayasuriya (1994) and Wik *et al* (2003) mainly focus on plant uncertainties that contain the two kinds of uncertainties mentioned above (structured and unmodelled). Zhao and Jayasuriya (1994) showed the following conclusions:

a.) The tracking error caused by plant uncertainties only relates to the desired trajectory, the plant uncertainties, and the feedback compensator;

b.) The FFC imposes a performance limitation on the tracking error;

c.) The steady-state tracking error will not be zero in the absence of a model of the desired trajectory in the control loop. This means that for zero-error tracking, the desired trajectory model has to be included into the control loop in the presence of model uncertainties.

Wik *et al* (2003) presented an approach to solve the uncertainty problem through optimization methods, and demonstrated that to provide the optimal performance the FFC and FBC have to be synthesized jointly. The parameter uncertainties therein are modelled using probability density functions. Then, the FFC and FBC are selected through expectation-value minimization of the performance index, which is a function of parameter uncertainties. In addition, the trade-off among performance, robustness, and controllability can be achieved by changing the constraints.

Devasia (2000; 2002) has shown conditions which relate to the issue of when to switch on or off the FFC, for the model affected by uncertainties (with respect to the worst-case tracking errors), as shown in Fig. 2.1. His results show that the system must satisfy the following two assumptions and conditions:

a.) **Assumption 1**: the nominal square plant $\boldsymbol{G}_0(s)$ has full normal rank.

b.) **Assumption 2**: the nominal system, the uncertainty, and the controller are such that the nominal and perturbed closed-loop systems are stable.

c.) **Condition 1**: the nominal plant $\boldsymbol{G}_0(s)$ has full rank at a given frequency ω. This means that $\boldsymbol{G}_0(s)$ does not have poles or transmission zeros at ω.

d.) **Condition 2**: uncertainty acceptability is satisfied:

$$\left\|\Delta(j\omega)\right\|_2 \le \boldsymbol{\delta}(j\omega) \le \frac{\left\|\boldsymbol{G}_0(j\omega)\right\|_2}{\boldsymbol{\kappa}_{\boldsymbol{G}_0}(j\omega)} \tag{8.1}$$

where $\boldsymbol{\kappa}_{\boldsymbol{G}_0}(j\omega)$ is the condition number of the nominal model $\boldsymbol{G}_0(s)$, $\Delta(j\omega)$ is the plant uncertainties defined by the difference of $\boldsymbol{G}_0(s)$ and $\boldsymbol{G}(s)$, and $\boldsymbol{\delta}(j\omega)$ is the bounded value for uncertainties.

Then the results shown by Devasia can be summarised in Table 8.1 with regard to the relationship between the sizes of uncertainties and the nominal model $\boldsymbol{G}_0(s)$ divided by its condition number. In Table 8.1, the term $\boldsymbol{E}_{(\boldsymbol{ff},*)}(j\omega)$ represents the worst-case tracking error with the inverse FFC and the term $\boldsymbol{E}_{(\boldsymbol{fb},*)}(j\omega)$ represents the worst-case tracking error with only the FBC. The symbol * (different forms of Δ) is a general symbol for uncertainties for the three different situations, represented in Table 8.1 respectively.

Table 8.1 Comparison of the tracking performance with or without the FFC

Size of uncertainties	Comparison of tracking performance
$\left\|\Delta(j\omega)\right\|_2 \le \frac{\left\|\boldsymbol{G}_0(j\omega)\right\|_2}{\boldsymbol{\kappa}_{\boldsymbol{G}_0}(j\omega)}$	For all controllers and any uncertainty $\Delta(j\omega)$ $\boldsymbol{E}_{(\boldsymbol{ff},\Delta)}(j\omega) \le \boldsymbol{E}_{(\boldsymbol{fb},\Delta)}(j\omega)$
$\frac{\left\|\boldsymbol{G}_0(j\omega)\right\|_2}{\boldsymbol{\kappa}_{\boldsymbol{G}_0}(j\omega)} < \left\|\Delta(j\omega)\right\|_2$	There exists a controller and an uncertainty $\bar{\Delta}(j\omega)$ $\boldsymbol{E}_{(\boldsymbol{ff},\bar{\Delta})}(j\omega) > \boldsymbol{E}_{(\boldsymbol{fb},\bar{\Delta})}(j\omega)$
$\left\|\boldsymbol{G}_0(j\omega)\right\|_2 < \left\|\Delta(j\omega)\right\|_2$	For any controller, there exists an uncertainty $\hat{\Delta}(j\omega)$ $\boldsymbol{E}_{(\boldsymbol{ff},\hat{\Delta})}(j\omega) > \boldsymbol{E}_{(\boldsymbol{fb},\hat{\Delta})}(j\omega)$

Although the above results quantitatively answer the question whether to use the FFC, practical systems always tend to have a relatively large uncertainty modelling error (such as

in the helicopter field where the rotor dynamics and interactions between rotors and fuselage often is a major source of uncertainty). Hence, Devasia (2002) introduced a modified inversion approach by first dividing the frequency domain into regions according to the degree of severity of the uncertainties. The approach only inverts the system in the part of the frequency domain where the uncertainty is sufficiently small. The results have shown that the modified approach can improve the tracking performance compared to the use of the FBC alone by introducing the FFC in the significant part of frequency range.

8.3 Design of the FFC for a linear Lynx-like helicopter model

In Chapter 7, the results have shown that the structure which only includes the FBC achieves good tracking performance for the standard ADS-33E manoeuvres but not for the severe manoeuvres. In this section, the research will be focused on that whether or not the tracking performance can be improved by introducing the FFC. The FFC is now included into Fig. 7.1 to form the final simulation model structure, as shown in Fig. 8.1.

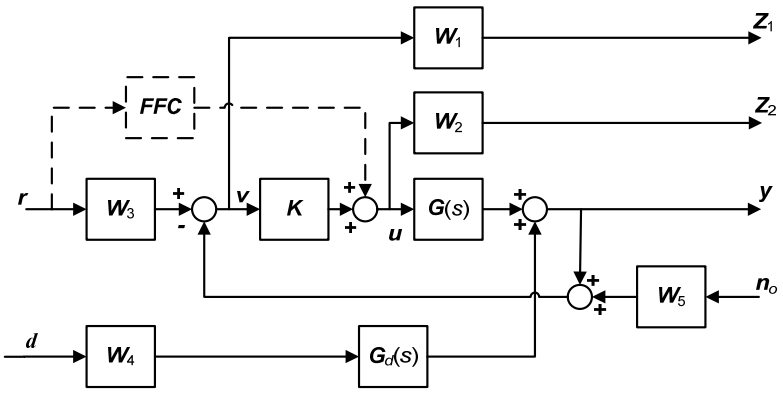

Fig. 8.1 Diagram of FFC+FBC system for the linear Lynx-like helicopter model

In this section, four groups of manoeuvres are considered. The first group is taken from the standard heave axis response (Walker & Postlethwaite, 1996) based on the latest version of ADS-33E-PRF (Anon, 2000). This group of manoeuvres is used to check the validity of the proposed method in which the FFC is designed using inverse simulation. The second group adopted the shape of the bob-up manoeuvre introduced in Chapter 2 for the heave velocity

channel. The third group represents a heading changing situation and the fourth group represents a different situation in terms of the degree of severity of manoeuvre. Among these four manoeuvres, the fourth group of manoeuvre is most demanding but may lack practically physical meaning. Taken together, the results from these four different cases provide useful insight and have facilitated the investigation of the influence of the FFC on the tracking performance.

8.3.1 Application to the first-group of manoeuvres

Since the available helicopter model is linearised around the hover situation, the desired vertical rate response is defined as having the qualitative appearance of a first-order lag with an additional pure delay, as shown in Eq. (3.12). The other three channels p, q, and $\dot{\Psi}$ are set to be zero in terms of their desired responses. The details can be found in Chapter 3 in this thesis. The results from this application are shown in Fig. 8.2 and Fig. 8.3.

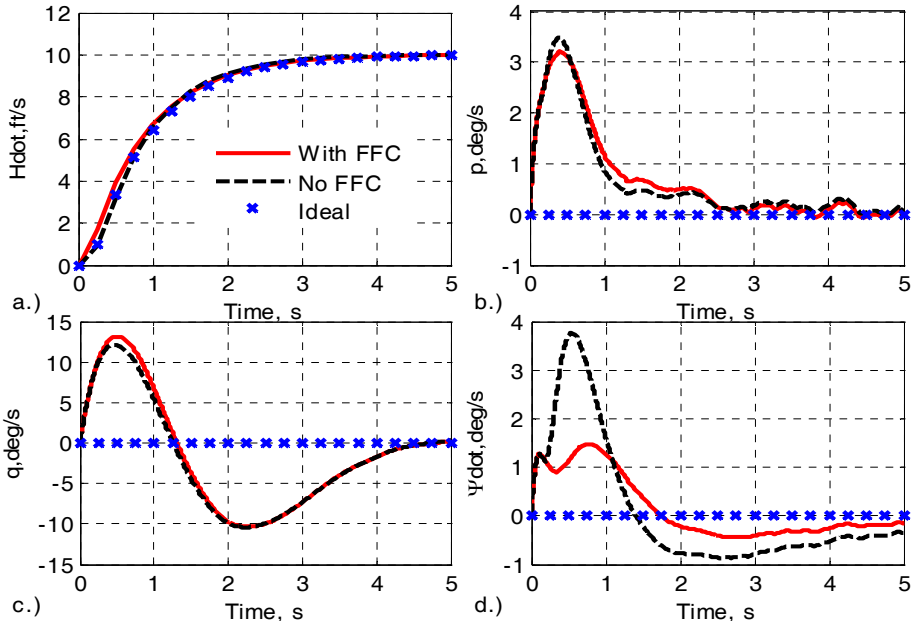

Fig. 8.2 Results from 2DOF system with and without FFC for the ADS-33E height-response manoeuvre with disturbances and measurement noise

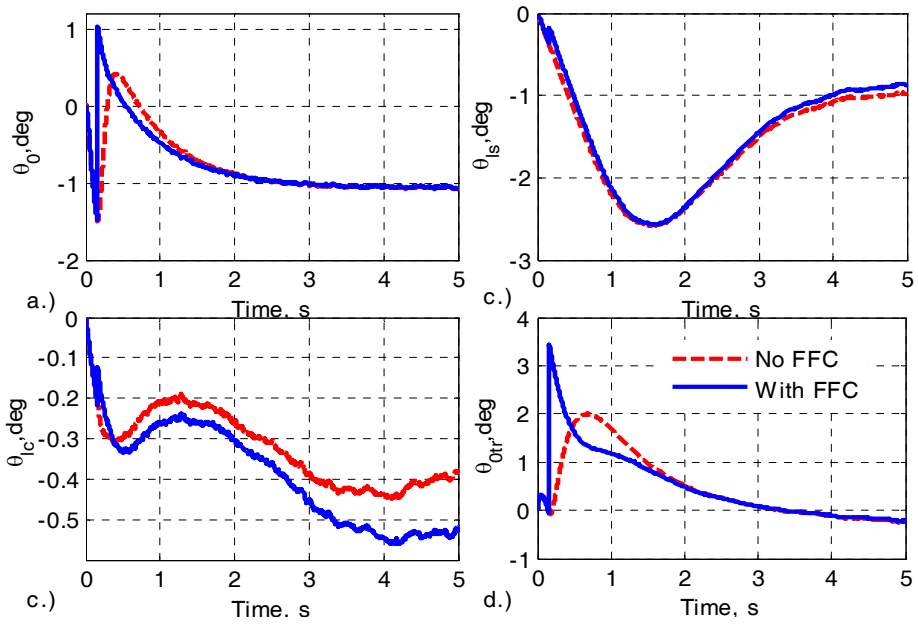

Fig. 8.3 Results showing control efforts from 2DOF system with and without FFC for the ADS-33E height-response manoeuvre with disturbances and measurement noise

Fig. 8.2 shows that for the simulations of standard manoeuvres the systems with and without the FFC provide almost the same tracking performance with disturbances and measurement noise. The simulation process is causal since no predefined information is required. This is one of the strengths of the proposed method over model inversion, as was mentioned in Chapter 3. The results in the channels representing $\dot{H}$, p, and q, with and without the FFC, are nearly the same. However, in the channel $\dot{\Psi}$ in Fig. 8.2d, the control structure with the FFC is apparently better than the one without the FFC.

Fig. 8.3 shows the comparison of control efforts from these different approaches. As shown in this figure, the control efforts in these four channels are nearly the same for these two control structures. The spikes shown in the channels θ_0 and θ_{0tr} in the initiating period result from the 'direct-control' effect of the FFC. Furthermore, this step input in the collective pitch θ_0 corresponds to an increase of the blade drag and consequently in the engine torque to accelerate the system to achieve the first-order step response in terms of heave velocity. Meanwhile, the step input in the tail rotor channel θ_{0tr} counteracts the effect of the main rotor to keep the heading stable. In addition, control inputs with FFC designed from the

inverse simulation procedure are bounded, regardless of the NMP characteristics of the vehicle.

Other investigations with increasing noise levels have also been carried out. The results are similar to those shown in Fig. 8.2 and Fig. 8.3 and therefore have not been included in this section. This property of robustness against measurement noise proves the effectiveness of the designed H_∞ controller once again. Furthermore, all results from the simulations show the validity of the proposed approach in terms of the proposed replacement of model inversion by inverse simulation for design of the FFC to improve the tracking performance.

8.3.2 Application to the second and third groups of manoeuvres

In this subsection, the second and third groups of manoeuvres are implemented to investigate the performance with the FFC designed from inverse simulation. The definition of the second group of manoeuvres follows rules which are similar to those for the first group except for the heave velocity channel. Instead of tracking the ADS-33E height-response manoeuvre, this channel in the second group corresponds to the velocity profile (Eq. 2.49) of the bob-up manoeuvre introduced in Chapter 2. The other three channels p, q, and $\dot{\Psi}$ are set to be zero in terms of their desired responses. The results from the second group are shown in Fig. 8.4 and Fig. 8.5. The third group, involves use of the step response of a standard second-order transfer function, as shown in Eq. (7.13), for the heading rate with $\xi = 0$ and $\omega_n = 1.5$ rad/s. The other three output channels are again set to be zero. The results from the second group are shown in Fig. 8.6 and Fig. 8.7.

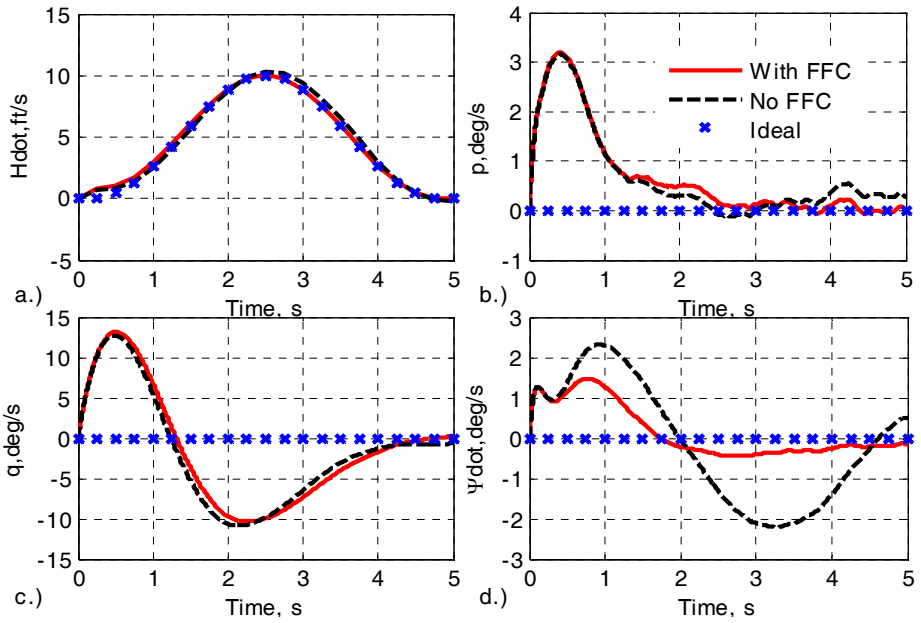

Fig. 8.4 Results from 2DOF system with and without FFC for the second group of manoeuvres with disturbances and measurement noise

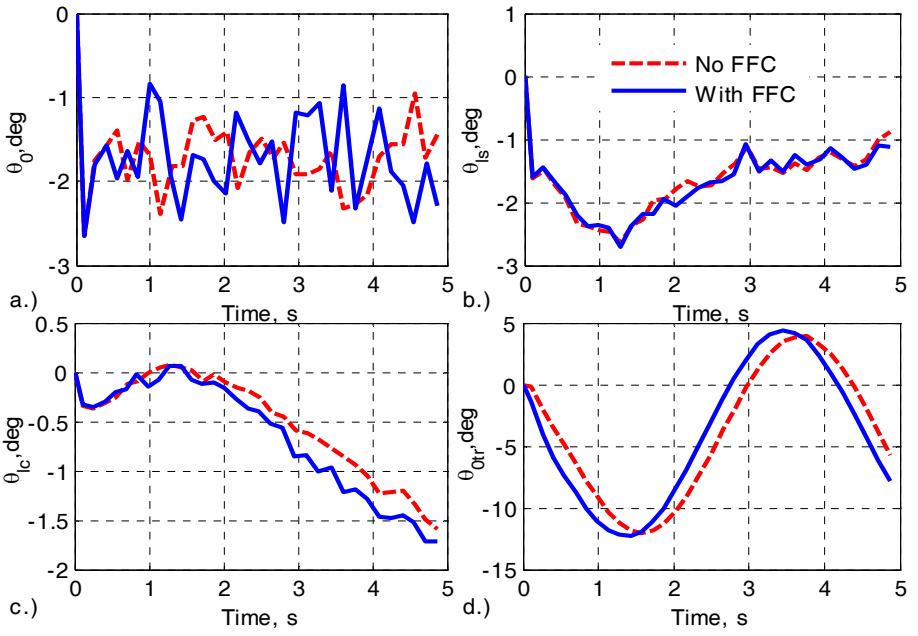

Fig. 8.5 Results showing control efforts from 2DOF system with and without FFC for the second group of manoeuvres with disturbances and measurement noise

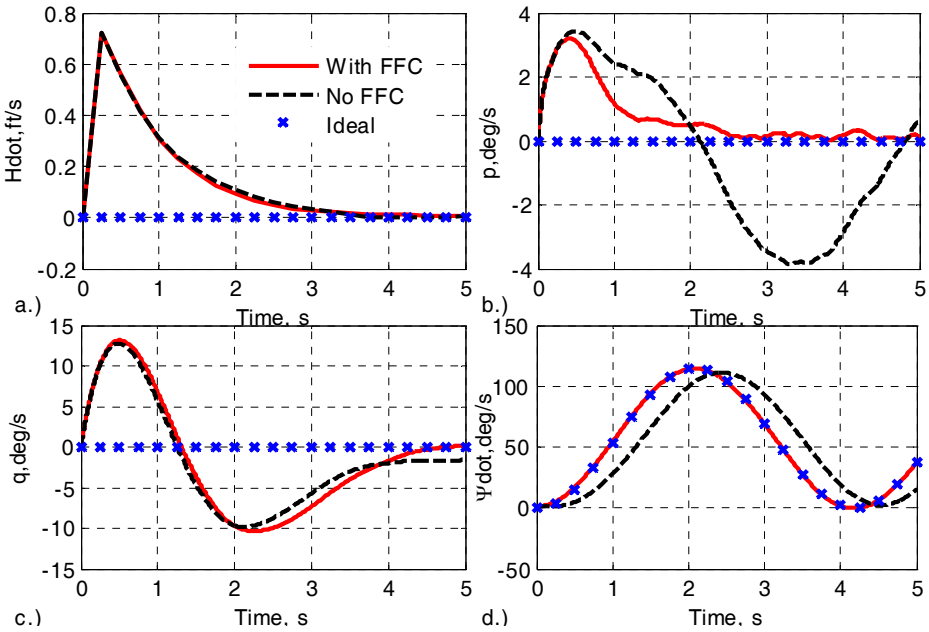

Fig. 8.6 Results from 2DOF system with FFC and without FFC for the third group of manoeuvres with disturbances and measurement noise

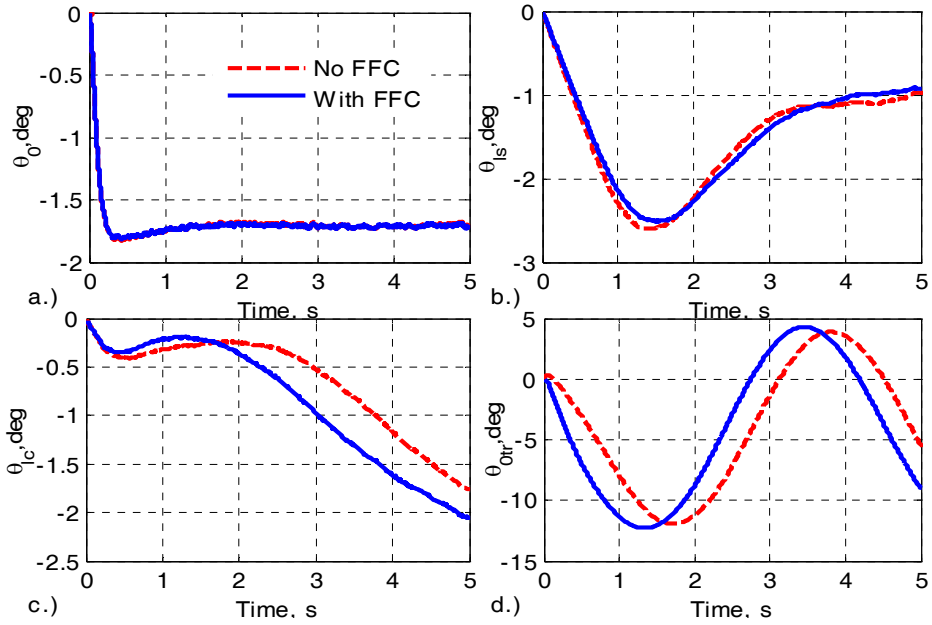

Fig. 8.7 Results showing control efforts from 2DOF system with and without FFC for the third group of manoeuvres with disturbances and measurement noise

Regardless of the different manoeuvres implemented, the results from the second group, as shown in Fig. 8.4 and Fig. 8.5, are similar to those from the first group, except in the case for the channel $\dot{\Psi}$ in Fig. 8.4d. In general the performance for the control structure with the FFC is far better than the one without the FFC. In addition, control inputs with the FFC designed from inverse simulation are again bounded regardless of the NMP characteristics of the vehicle as well as the different manoeuvre being implemented.

The tracking performance from the third group is a little different from that shown in Fig. 8.2 and Fig. 8.4. Compared with the situation without the FFC, the case when the FFC is included achieves better tracking performance in the channels p (Fig. 8.6b) and $\dot{\Psi}$ (Fig. 8.6d). The difference in the heading tracking between these two control structures also is shown in the discrepancy of control efforts in the tail rotor channel, as shown in Fig. 8.7d. In addition, the *sinusoidal* shape of the tail rotor pitch follows the heading manoeuvre illustrated in Fig. 8.6d.

8.3.3 Application to the fourth group of manoeuvres

In this part, the fourth group, which consists of more demanding manoeuvres, is selected to compare the tracking performance with or without the FFC. The specification of this group can refer to the definition of the demanding manoeuvres in Section 7.4. The results from simulations with the measurement noise and external disturbances are presented in Fig 8.8. However, results from other investigations with increasing noise levels are similar to these and therefore are not presented here. In addition, control inputs are ignored since these manoeuvres lack practical physical meaning and the manoeuvres could not be achieved if control input constraints were to be considered.

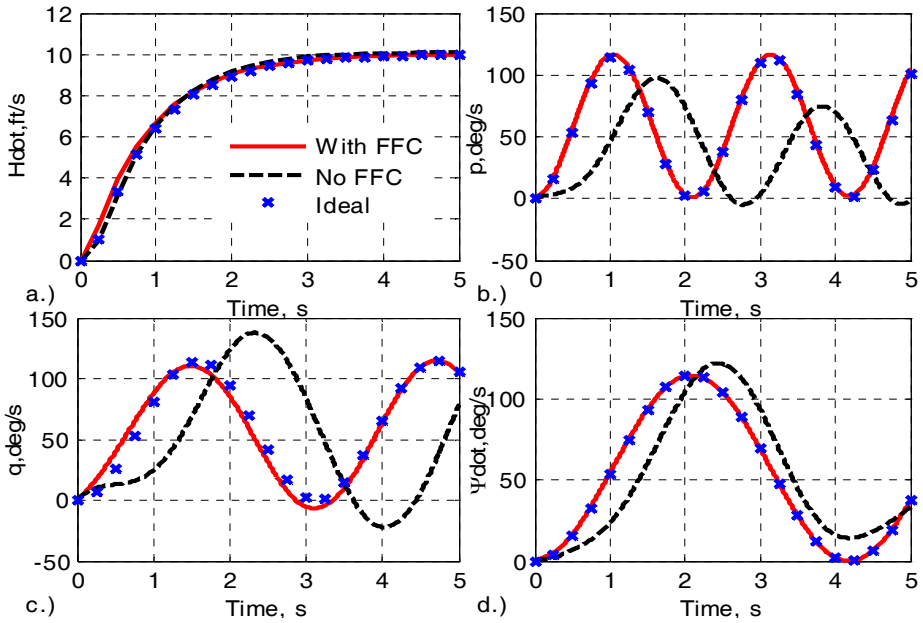

Fig. 8.8 Results from 2DOF system with FFC and without FFC for the fourth group of manoeuvres with disturbances and measurement noise

The tracking performance shown in Fig. 8.8 is quite different from that shown in the above three cases for the less demanding manoeuvres. From these figures, it can be observed that the structure with the FFC always provides good results, which are far better than the results without the FFC for the severe manoeuvres. In Fig. 8.8, the structure without the FFC achieves good tracking only in the heave rate channel (Fig. 8.8a) and the results of the other three channels are unsatisfactory. In addition, further investigations suggest that the use of smaller discretization intervals will lead to more accurate tracking with the FFC. This suggests that perfect tracking performance is achievable within the simulation environment. However, the same results cannot be found for the structure without the FFC.

The different performances from these two types of manoeuvres (demanding or not-demanding) is probably due to one of the advantages of the FFC in that it can relate the system response directly to commands by providing a 'direct-control' channel. In addition, it is known that, generally, the smaller the discretization interval, the more accurate will be the control inputs obtained from inverse simulation (Hess *et al.*, 1991). In terms of the control effort comparison, results are broadly similar for both structures, as shown in Figs.

8.3 and 8.5 and 8.7. All these results demonstrate the stability and applicability of the algorithm for inverse simulation and the validity of the proposed approach for NMP systems for various kinds of manoeuvres. In addition, the inclusion of the FFC can improve the tracking performance.

8.4 Design of the FFC for a nonlinear Container ship model

In Chapter 7, the performances of the trajectory tracking and the RRS, which are the two control objectives to be accomplished, with the FBCs designed from the LQ technique and the H_∞ algorithm have been compared for the Son and Nomoto full nonlinear Container ship model. The results show that better performance is obtained when the H_∞ algorithm is implemented. Now the final simulation structure is similar to Fig. 8.1 but with the addition of a prefilter $\boldsymbol{F}_i$, as shown in Fig. 8.9.

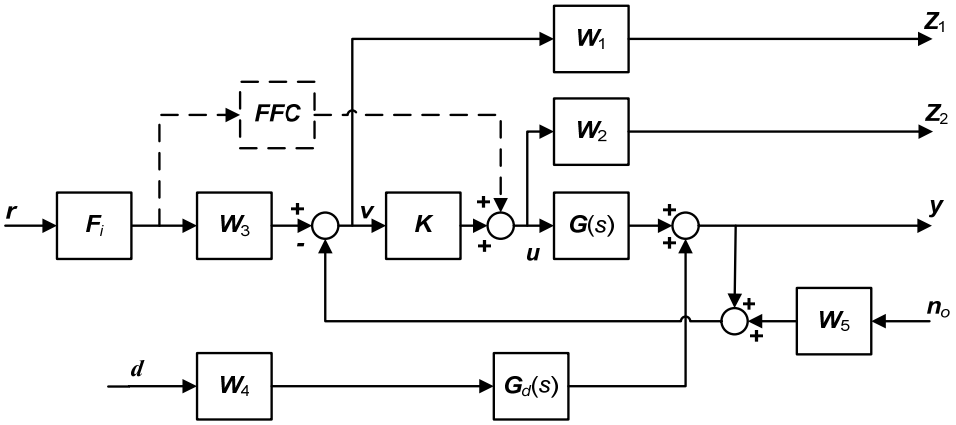

Fig. 8.9 Diagram of FFC+FBC system for the nonlinear Container ship

In this diagram, the prefilter $\boldsymbol{F}_i$, which is a standard second-order system, as shown in Eq. (7.13), is added to avoid the numerical problems resulting from large step inputs. Two kinds of FFCs, linear and nonlinear, have been designed. For the design of the linear FFC, the one input (heading angle) and one output (rudder angle) model similar to the one shown in Eq. (7.14), but without the disturbance, is considered as the benchmark for design of the linear FFC using inverse simulation, since ship steering is the more important factor. The

nonlinear model used to design the nonlinear FFC is similar to the model adopted to validate the design of the FBCs with the control constraints, but without the disturbance part. In addition, the final simulation is run on a full-scale nonlinear Container ship model with a forward surge speed U = 7.3 m/s by replacing $\boldsymbol{G}(s)$ in Fig. 8.9 by the equivalent nonlinear description.

Three cases of manoeuvres generated by changing the coefficients in the prefilter with $\omega_n = 0.015$ rad/s , $\omega_n = 0.05$ rad/s , and $\omega_n = 0.1$ rad/s are investigated in this section. In addition, for all cases, ξ is selected to have the same value of 0.9. In addition to three kinds of manoeuvres, the investigations are also performed for three different situations. These are as follows: (a) simulations without the FFC, which have been discussed in detail in Chapter 7; (b) simulations with the linear FFC; and (c) simulations with the nonlinear FFC.

For the linear FFC, the inverse simulation technique based on the integration process (Hess *et al*., 1991) is implemented for its simplicity and fast convergence. In addition, the disturbance model in the linear model shown in Eq. (7.14) has been ignored here. Furthermore, there are no input constraints in the adopted linear model. All these can help to guarantee good quality results from the inverse simulation approach based on the integration process.

As far as the nonlinear model used to design the nonlinear FFC is concerned, the difference between this and the nonlinear simulation benchmark model lies in the fact that the disturbance model is ignored and input constraints are included. For the methodology to design the nonlinear FFC, the constrained derivative-free inverse simulation approach based on the NM algorithm, developed in Chapter 6, is adopted due to the failure of the integration-based inverse simulation approach for application to this kind of situation in which input constraints are included. This represents the motivation for development of the NM technique for inverse simulation, which was discussed at length in Chapter 6.

The results from the simulation of the first case are shown in Fig. 8.10 and Fig. 8.11 and the RMS values in Table 8.2 facilitate quantitative comparison of the performance with or without the FFC. The results from the second case are shown in Fig. 8.12 and Fig. 8.13 and also in Table 8.3, and those from the third case are shown in Fig. 8.14 and Fig. 8.15 and also in Table 8.4. In addition, during the simulation period, the feedback for the channels p and Φ are switched on for the period of time from 300 s to 500 s and are switched off at all other times, in order to show the performance in terms of roll-moment reduction.

Table 8.2 Comparison of RMS values from 2DOF system with the nonlinear Container ship model (ω_n = 0.015 rad/s). Band: the time period from 300s to 500 s; total: the whole period

Structure	Φ(Total, deg)	Φ(Band, deg)	p(Total, deg/s)	p(Band, deg/s)	u(Total, deg)
Without FFC	2.45	1.84	0.603	0.490	0.838
With linear FFC	3.78	1.77	0.879	0.407	1.22
With nonlinear FFC	2.86	1.57	0.683	0.438	0.886

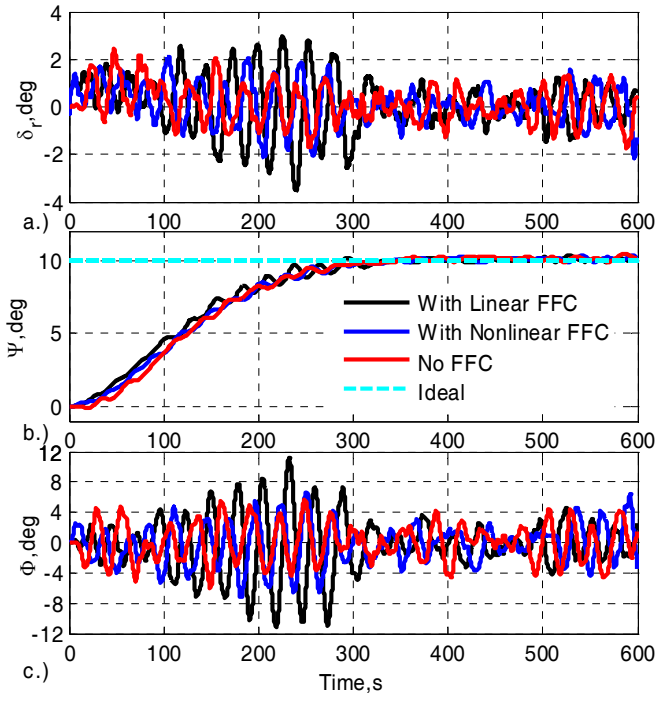

Fig. 8.10 Results from 2DOF system with the nonlinear Container ship model (U = 7.3 m/s, ω_n = 0.015 rad/s)

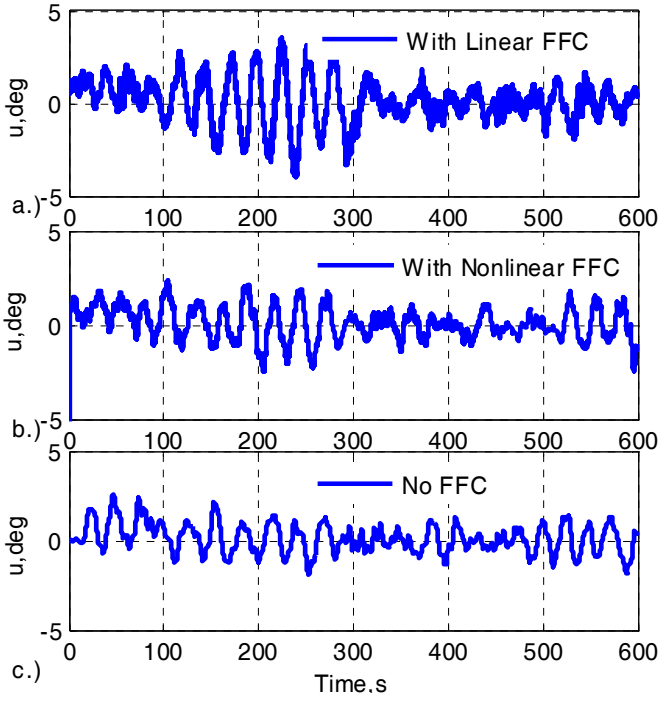

Fig. 8.11 Inputs from 2DOF system with the nonlinear Container ship model (U = 7.3 m/s, ω_n= 0.015 rad/s)

Table 8.3 Comparison of RMS values from 2DOF system with the nonlinear Container ship model (ω_n = 0.05 rad/s). Band: the time period from 300s to 500 s; total: the whole period

Structure	Φ(Total, deg)	Φ(Band, deg)	p(Total, deg/s)	p(Band, deg/s)	u(Total, deg)
Without FFC	3.56	2.44	0.815	0.593	1.50
With linear FFC	2.26	1.91	0.540	0.476	1.46
With nonlinear FFC	2.91	1.73	0.693	0.478	1.35

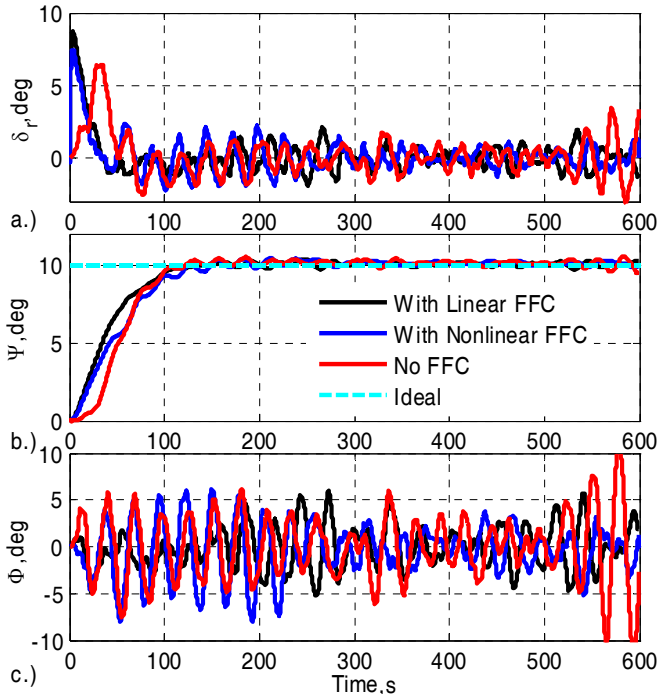

Fig. 8.12 Results from 2DOF system with the nonlinear Container ship model (U = 7.3 m/s, ω_n = 0.05 rad/s)

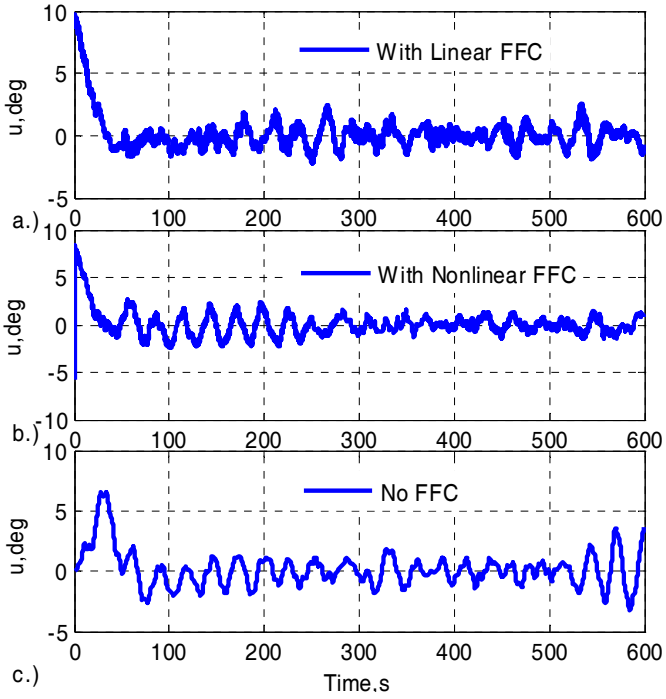

Fig. 8.13 Inputs from 2DOF system with the nonlinear Container ship model (U = 7.3 m/s, ω_n = 0.05 rad/s)

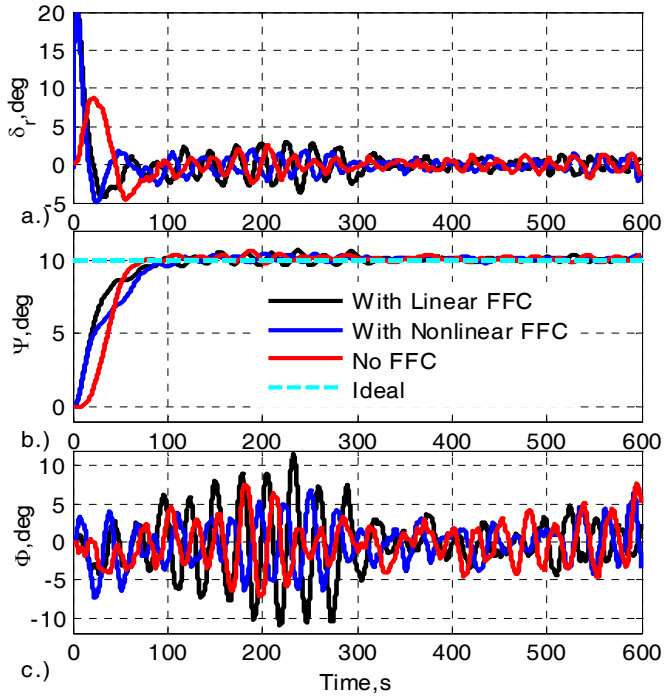

Fig. 8.14 Results from 2DOF system with the nonlinear Container ship model (U = 7.3 m/s, ω_n = 0.1 rad/s)

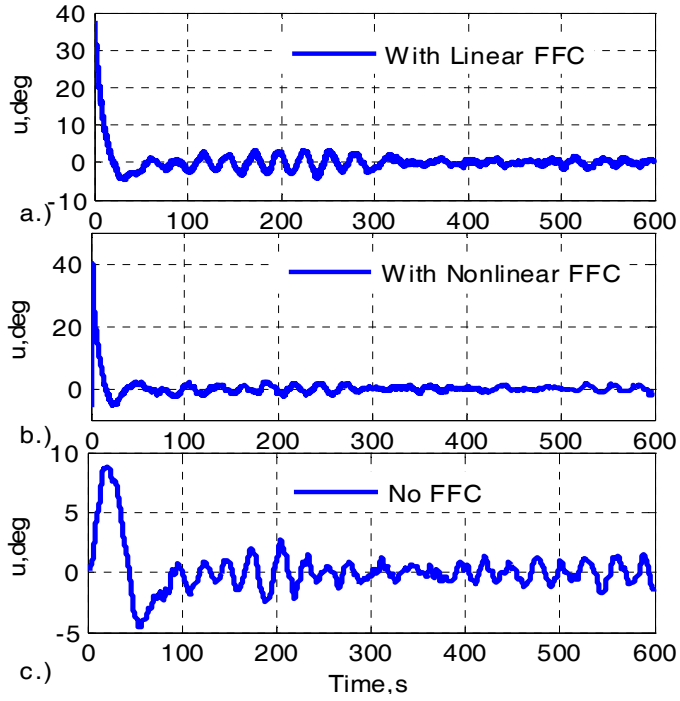

Fig. 8.15 Inputs from 2DOF system with the nonlinear Container ship model (U = 7.3 m/s, ω_n = 0.1 rad/s)

Table 8.4 Comparison of RMS values from 2DOF system with the nonlinear Container ship model (ω_n = 0.1 rad/s). Band: the time period from 300 s to 500 s; total: the whole period

Structure	Φ(Total, deg)	Φ(Band, deg)	*p*(Total, deg/s)	*p*(Band, deg/s)	*u*(Total, deg)
Without FFC	2.84	2.16	0.630	0.501	1.95
With linear FFC	4.01	1.84	0.927	0.421	3.46
With nonlinear FFC	2.73	1.60	0.654	0.407	3.50

From the graphs and the RMS values shown in the tables above, the structures of the *K*/*KS* FBC systems with or without the FFC both show good performance against the measurement noise and disturbances. The word 'Band' in the tables signifies the period of time from 300 s to 500 s and 'Total' represents the whole time period.

For the case of the gentle manoeuvre with ω_n = 0.015 rad/s, these values in Table 8.2 show that the roll angle (Φ) has been reduced by 0.6 deg and the roll rate (*p*) by 0.12 deg/s when the roll-moment feedback is switched on for the case without the FFC. When the linear FFC is introduced, the performance improvement in the band period is slightly better and is further reduced down to 1.77 deg (Φ) and 0.407 deg/s (*p*), as shown in Table 2. However, the performance over the whole time period is worse (3.78 deg) as can be observed in Fig. 8.10c in which some roll angles are significantly larger. If the nonlinear FFC is included, the results correspond to those for the linear FFC case but the band roll angle reduces further to 1.57 deg. However, in contrast to the decrease of the roll angle, the value of the band roll rate is increased. This is due to the fact that the relationship between the roll angle and the roll rate is not a simple integration process and involves the different reference systems.

In addition to the roll-reduction achievement, both structures achieve nearly perfect heading tracking shown in Fig. 8.10b, although the RMS values of the rudder angles in the whole time period for the case with the FFC are slightly larger than the ones without the FFC, as presented in Table 8.2. The input values $\boldsymbol{u}$ in Table 8.2 may be different from the values of δ_r shown in Fig. 8.10a in that the values $\boldsymbol{u}$, which are the control effort summation between the FBC and the FFC, are the input for the nonlinear model and in fact it is equal to the

term δ_c in Fig. 6.4. The δ_r values in Fig. 8.10 involve the actual rudder deflections and are equal to the constrained values of $\boldsymbol{u}$. Among the $\boldsymbol{u}$ values in Table 8.2, the ones with the FFC are larger but are still within the constraint limits. This is due to the fact that the $\boldsymbol{u}$ values for the case with the FBC involve contributions from both the FFC and FBC.

The analyses for the second case can follow in the same way as for the first case and therefore are ignored here. When the manoeuvres become more severe by increasing the ω_n value for the prefilter $\boldsymbol{F}_i$ to 0.1 rad/s, both structures achieve nearly perfect heading tracking and good roll-reduction as well, as shown in Fig. 8.14 and Table 8.4. However, Fig. 8.15 shows that the rudder angles $\boldsymbol{u}$ with the FFC are quite large (reaching to around 40 deg) in the first few seconds but decrease sharply and the system then operates with very small magnitudes of input. The initial large amplitudes with the FFC result, in part at least, from the fast tracking requirement for this new manoeuvre. Therefore, compared with the previous gentle manoeuvre (ω_n = 0.015, 0.005 rad/s), the larger rudder deflections are required to provide the necessary control efforts. However, since the maximum values of the input amplitude without the FFC are only around 8 deg, the large amplitudes required with the FFC are primarily due to the 'direct-control' function from the control effort of the FFC as mentioned earlier. As a result, in this case, the rudder is working under saturation conditions, as illustrated in Fig. 8.14a and shown also by the relatively large $\boldsymbol{u}$ values in Table 8.3.

As been analysed in Chapter 7, the good properties of the *K/KS* controller avoid the need to introduce the extra complexity of dealing with measurement noise in this case. The same analysis can be applied in terms of disturbance rejection. All these results prove the effectiveness of the proposed structure with the FFC designed from the inverse simulation for these applications.

8.5 Summary

The 2DOF structure based on inverse simulation for the FFC has been designed and combined with an effective H_∞ controller as the FBC. The results from an eighth-order

linearised NMP helicopter model have shown the validity and effectiveness of the approach. The implemented structure always achieves good tracking performance regardless of the manoeuvre adopted. The bounded results from the causal simulation process also prove the feasibility of inverse simulation for replacement of more complex model-inversion techniques in such situations. In addition, the magnitudes of the control efforts involved in the new approach are nearly the same as those without the FFC.

These advantages are further elaborated in case studies involving an application to the Son and Nomoto nonlinear Container ship model. The results show good performance against measurement noise and disturbances as well as good rudder-roll reduction during periods of time when the feedback of the channels p and Φ is switched on. These investigations prove the effectiveness of the approach and confirm that inverse simulation may be used successfully in the design of the FFC for a complex nonlinear model. The results with the nonlinear FFC show the smallest roll angle and it is suggested that this is achieved from the additional information contained in the feedforward channel. For gentle manoeuvres, the inputs from the combination of the FFC and FBC are within the constraints. However, one apparent shortcoming lies in the fact that large control efforts are required in the first few seconds of a more severe manoeuvre due to ‘direct control’ components from the FFC. This can cause the rudder to operate at the saturation limit for some time. The problem may be overcome by adjusting the weighting function $\boldsymbol{W}_2$ to limit the control efforts.

Chapter 9

An Investigation of Helicopter Ship Landing Using Inverse Simulation

Contents

Since the first successful landing by a helicopter on board a merchant ship (the *Daghestan*) in 1943, the tasks performed by helicopters quite often involve marine operations such as civil air-sea rescue and military actions. Therefore, much research has been carried out focussing on the issue of helicopter shipboard landing. The aim of this chapter is to simulate a helicopter ship landing mission onto a deck using the inverse simulation technique and to investigate control based upon the combined 2DOF feedback and feedforward controller structure described in earlier chapters. Aspects of the work presented here are based on the contribution made by Thomson, Coton and Galbraith (2005).

9.1 Manoeuvre definition

As mentioned in Chapter 2, inverse simulation is a methodology that can determine the control actions required by the model of a vehicle to track perfectly a specified manoeuvre. Therefore, the first step of the inverse simulation procedure is to define an ideal trajectory that suits the practical situation. In his thesis McGeoch (2005) breaks the ship landing process into six individual manoeuvres: cruising flight, descent and deceleration, a 45^{o}

approach, alignment, lateral reposition over the deck, and final touchdown, by following the mission task elements within ADS-33 (Anon, 2000). The landing procedure defined in this way represents a typical and practical shipboard recovery process but this multi-phase approach is complicated and is not feasible to implement for inverse simulation. Therefore, a simpler version of the manoeuvre definition needs to be found and the procedure described by Thomson *et al* (2005) has been adopted for this work. In this approach the three Earth-fixed reference coordinates with the additional heading angle are used to define the four elements of the manoeuvre, as introduced in Chapter 4. The manoeuvre scheme for helicopter ship landing is shown diagrammatically as follows in Fig. 9.1.

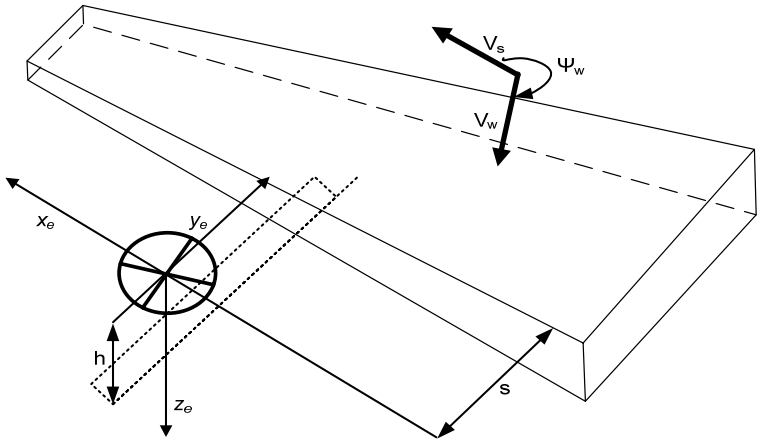

Fig. 9.1 Diagram showing definitions of relevant variables for the helicopter ship landing situation

In Fig. 9.1, x_e, y_e, and z_e are the set of coordinates of the helicopter within the Earth-fixed reference system. The direction of the x_e axis corresponds to the heading of the ship which is cruising at a speed of V_s. Ψ_w refers to the angle between the direction of the ship heading and the wind. V_w represents the speed of the wind. The centre of gravity of the helicopter is consistent with the location of the origin of the Earth-fixed reference system. At the initial stage, the helicopter is located at a height h above the ship and is at a distance s on its port side. The whole landing process starts from this initial state and can be divided into three phases:

a. The lateral motion phase

The velocity profile of this lateral motion phase is quite similar to the bob-up manoeuvre defined in Chapter 2 but involves lateral rather than vertical motion. The manoeuvre is initiated with the helicopter in a hover situation relative to the ship and the helicopter then accelerates sideways until the lateral velocity reaches a peak value $\dot{y}_{\max}$ at the time point $t_1/2$ (where t_1 is the time to complete the lateral motion). After this phase of the manoeuvre, the vehicle decelerates to a hover directly above the vessel. The total distance covered in this period is s. Therefore, the polynomial equation used to describe the bob-up manoeuvre in Chapter 2 can still apply to the case being discussed here.

$$\dot{y}_e(t) = \dot{y}_{\max}[-64(\frac{t}{t_1})^6 + 192(\frac{t}{t_1})^5 - 192(\frac{t}{t_1})^4 + 64(\frac{t}{t_1})^3] \tag{9.1}$$

where the value of the parameter t_1 can be calculated in a similar fashion to Eq. (2.47). The other three channels are defined as follows:

$$\dot{x}_e(t) = V_s\,; \quad \dot{z}_e(t) = 0 \;; \quad \Psi(t) = 0 \tag{9.2}$$

where the heading angle (Ψ) is assumed to be zero. This is sufficient for the still air condition but if the effects of wind disturbances have to be taken into consideration, the selection of this angle depends on the strategy adopted by pilot.

b. The stabilisation phase

In practical situations, it usually takes several seconds for the helicopter to stabilise over the deck before attempting a vertical landing. Therefore, this extra period has been taken into consideration in designing the manoeuvre implemented in this part of the work. The time consumed in this stabilisation phase is assumed to be t_2. The four channels are selected as follows:

$$\dot{x}_e(t) = V_s\,; \quad \dot{y}_e(t) = 0 \;; \quad \dot{z}_e(t) = 0 \;; \quad \Psi(t) = 0 \tag{9.3}$$

c. The vertical landing phase

The velocity profile of this phase is similar to that in the first lateral motion phase, although it involves a different axis. Thus, the polynomial equation is adopted as follows:

$$\dot{z}_e(t) = \dot{z}_{\max}[-64(\frac{t^*}{t_3})^6 + 192(\frac{t^*}{t_3})^5 - 192(\frac{t^*}{t_3})^4 + 64(\frac{t^*}{t_3})^3] \tag{9.4}$$

where $t^* = t - (t_1 + t_2)$ and t_3 is the total descent time. This quantity can be found using Eq. (2.47) likewise. The other three channels are defined as follows:

$$\dot{x}_e(t) = V_s; \quad \dot{y}_e(t) = 0 \; ; \quad \Psi(t) = 0 \tag{9.5}$$

9.2 Inverse simulation of the ship landing process

In this section two cases, with and without atmospheric disturbances, are investigated during the ship landing process. The values selected for the parameters of the manoeuvre scheme shown in Fig. 9.1 are as follows:

$$\begin{aligned} &V_s = 10 \text{ knots}; \; s = 20 \text{ m}; \; \dot{y}_{max} = 10 \text{ knots} \\ &h = 10 \text{ m}; \; \dot{z}_{max} = 5 \text{ m/s}; \; t_s = 1 \text{ s}; \end{aligned} \tag{9.6}$$

The final manoeuvres describing the landing process are shown as follows:

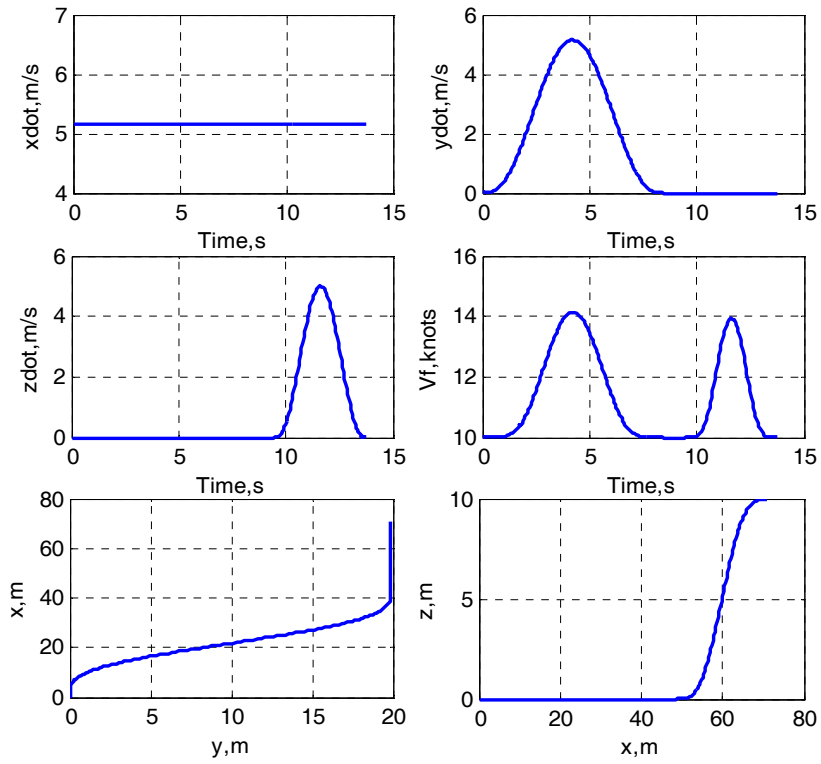

Fig. 9.2 The ideal trajectories for inverse simulation for the nonlinear Lynx helicopter model

9.2.1 Ship landing in still air

Based on the above manoeuvre definition and configuration, inverse simulation has been investigated for the nonlinear Lynx helicopter model (Bagiev, 2006) in still air. The results from the simulation process are illustrated from Figs. 9.3 to 9.6.

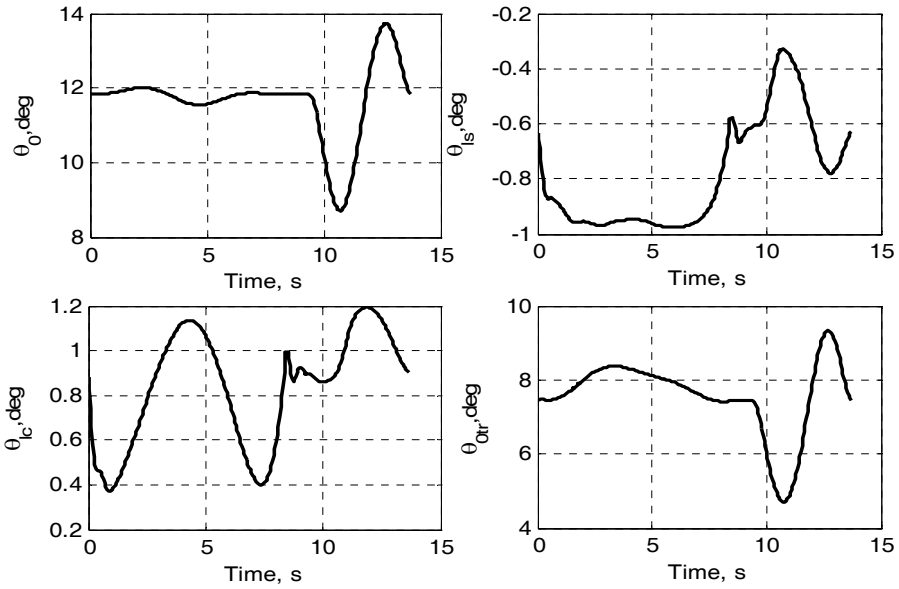

Fig. 9.3 Control efforts from ship landing process for the nonlinear Lynx helicopter model (Δt = 0.05 s, V =10 knots)

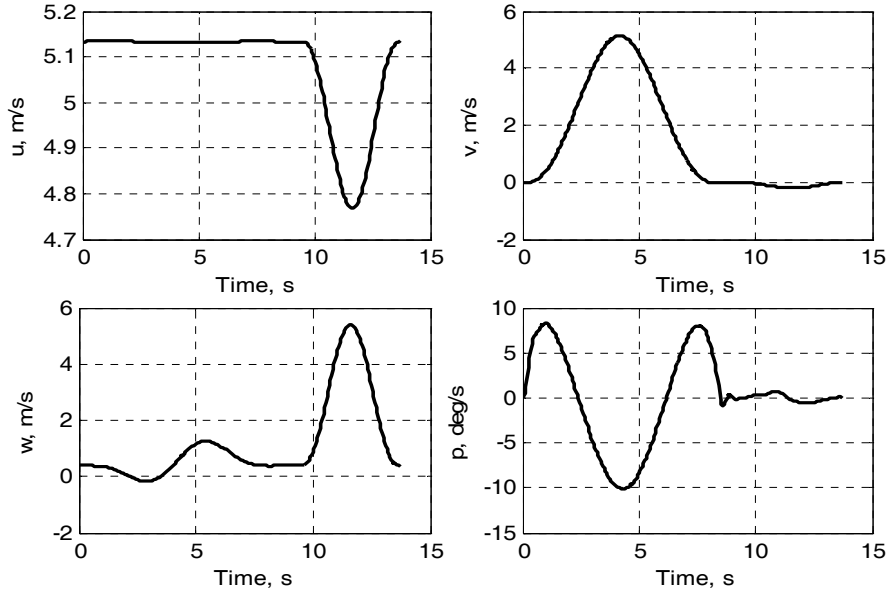

Fig. 9.4 State variable values from ship landing process for the nonlinear Lynx helicopter model (Δt = 0.05 s, V =10 knots)

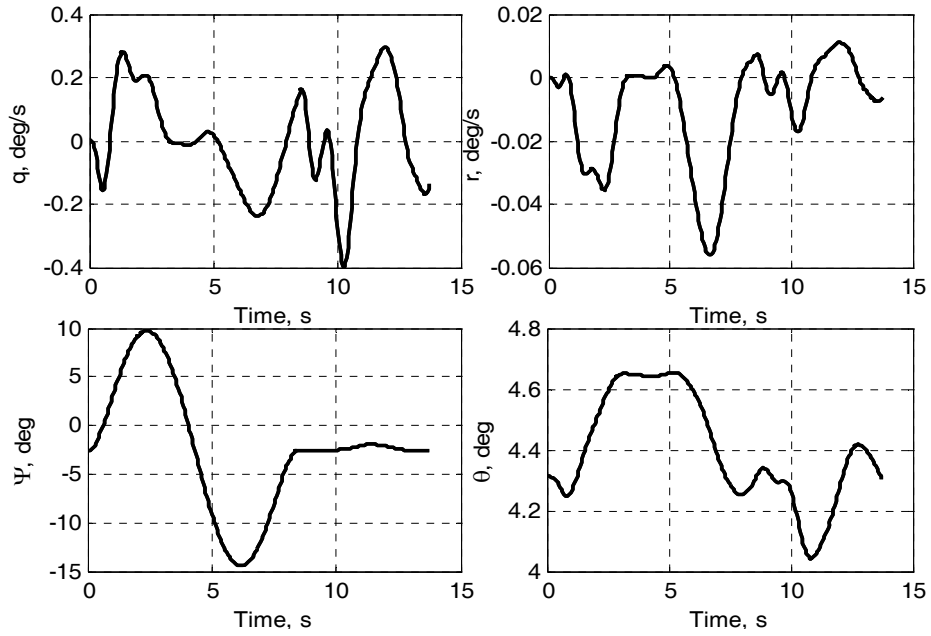

Fig. 9.5 State variable values from ship landing process for the nonlinear Lynx helicopter model (Δt = 0.05 s, V =10 knots)

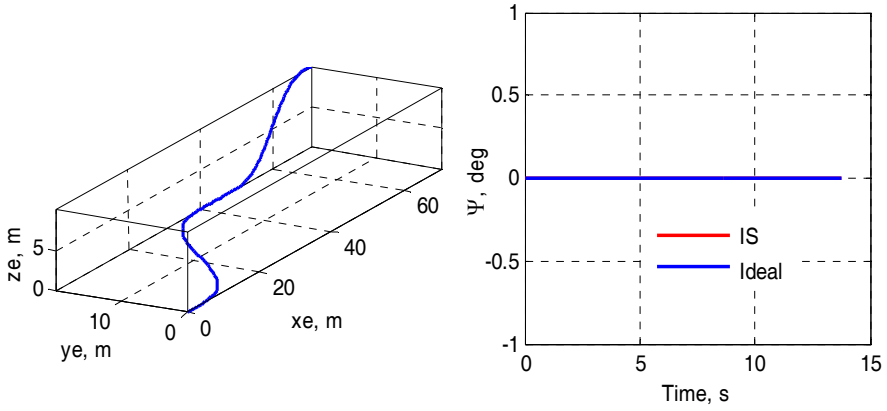

Fig. 9.6 Trajectory comparisons between ideal values and the ones from inverse simulation for ship landing process for the nonlinear Lynx helicopter model (Δt = 0.05 s, V = 10 knots)

Fig. 9.3 shows that the lateral cyclic control channel (θ_{1c}) initiates a step input and then a step in the opposite direction to achieve the sideways motion which lasts around 8 seconds. During this phase, the input through the collective channel (θ_{1c}) remains almost constant since the height of the helicopter has to be maintained. The longitudinal cyclic control input (θ_{1s}) also remains almost constant. The slight bump shown in the tail rotor channel (θ_{0tr}) is

used to balance the torque generated by the sideways motion. After the one-second stabilisation, this channel initiates a pulse to descend to the deck and generates a *sinusoidal* shape of control input time histories. A similar shape can be observed in the other channels as well, in order to keep the constant heading and roll and pitch stability.

Figs. 9.4 and 9.5 show the results in terms of the state variables obtained from the inverse simulation process. An analysis similar to that presented for the control channels can be applied to these variables. Fig. 9.6 shows comparisons between the ideal trajectories and the values from the forward simulation with the calculated control efforts. This proves the successful implementation of the inverse simulation process for the ship landing situation in the absence of wind disturbances.

9.2.2 Ship landing with wind disturbance

In this section, atmospheric disturbances are included in the equations of the helicopter mathematical model. In practice, the helicopter model becomes significantly more complicated and has time-varying characteristics if the wind disturbances are considered. Therefore, a straightforward method has been adopted to avoid this complexity in this work. First, the velocity of the helicopter with respect to the wind $(\boldsymbol{V}_{h,a})_e$ in the Earth-fixed reference system is obtained from the speed of the helicopter and the speed of the wind both with respect to the ground $(\boldsymbol{V}_{h,g}, \boldsymbol{V}_{a,g})_e$; secondly, $(\boldsymbol{V}_{h,a})_e$ is transformed into the values $(\boldsymbol{V}_{h,a})_b$ in body axes. Here the subscript b represents the body reference system; finally, the values $(\boldsymbol{V}_{h,a})_b$ are added to the corresponding variables in the equations of motion.

Four types of wind situations, as shown in following list, are investigated in this chapter. This choice is linked to the convenience of the specification of the wind velocity under these situations:

$\Psi_w = 0^0$ Tail wind

$\Psi_w = 90^0$ Port wind

$\Psi_w = 180^0$ Head wind

$\Psi_w = 270^0$ Starboard wind

The results based on these four wind situations are shown in Figs. 9.7 and 9.8, respectively.

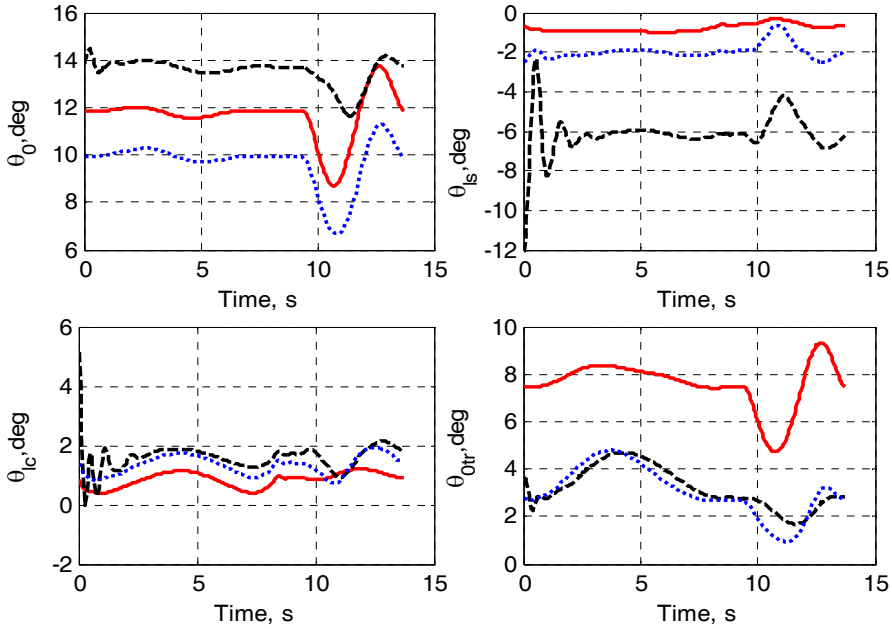

Fig. 9.7 Control efforts from inverse simulation with and without the wind disturbance ——still air; ----- 40 knots, headwind; ·····40 knots, tailwind

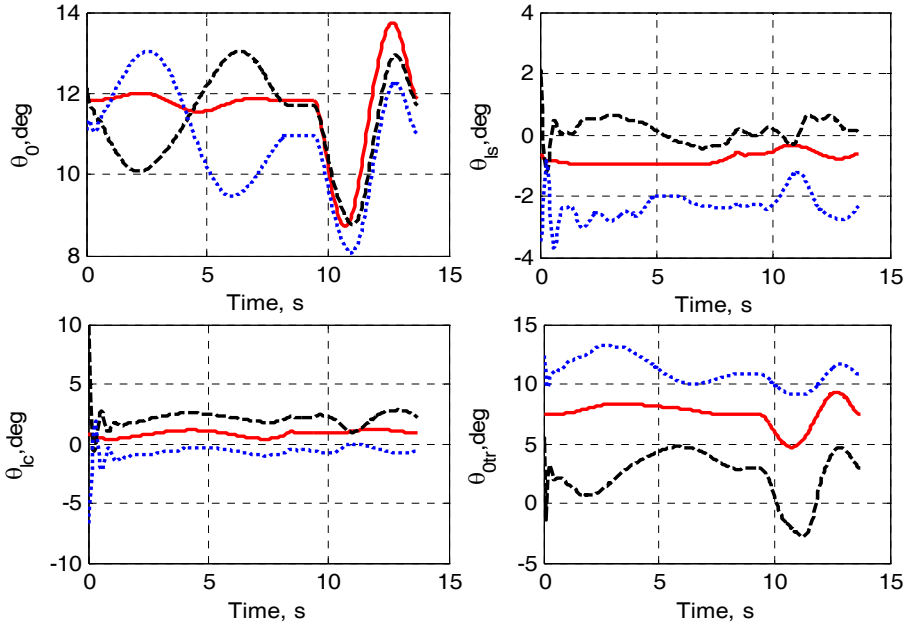

Fig. 9.8 Control efforts from inverse simulation with and without the wind disturbance ——still air; ----- 40 knots, port wind; ····· 40 knots, starboard wind

Fig. 9.7 shows that the four control efforts have similar shapes but different amplitudes. The results are just offset from those obtained in the still air conditions. This illustrates that the control strategies for landing on a deck might be similar for the situations involving headwind and tailwind. A similar phenomenon can be observed in the channels for the longitudinal, the lateral, and the tail rotor inputs in Fig. 9.8, although the collective case differs there. The collective channel requires a large control effort to overcome the addition contribution from the starboard wind for the sideways motion. In contrast, a lower control effort is demanded for the case of the port wind. However, for the both cases a similar shape is found in the vertical descending stage.

9.3 Investigation of ship landing with the 2DOF structure

This section focuses on two cases of applications of the ship landing manoeuvre using the 2DOF control scheme shown in Fig. 8.1. The first case involves the same Lynx-like linear helicopter model implemented in Section 8.3. The second case is based on a linear Lynx helicopter model which is linearised from the nonlinear helicopter model at a forward speed of 10 knots (Bagiev, 2006).

9.3.1 Ship landing based on the Lynx-like helicopter model

For this case, all the experimental configurations (except the ideal manoeuvres) are exactly the same as the ones investigated in Section 7.4 and Section 8.3, including the presence of disturbances and measurement noise. The manoeuvres adopted are derived from the inverse simulation process of the ship landing for the nonlinear helicopter model. The final simulation results are shown in Fig. 9.9 and Fig. 9.10.

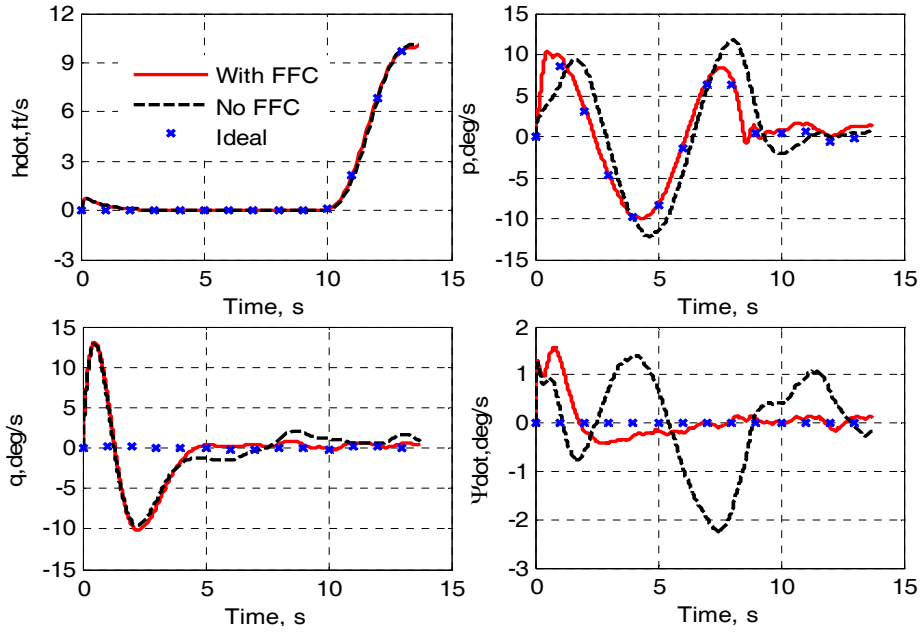

Fig. 9.9 Results from 2DOF with FFC and without FFC for ship landing with disturbance and measurement noise (Lynx-like helicopter)

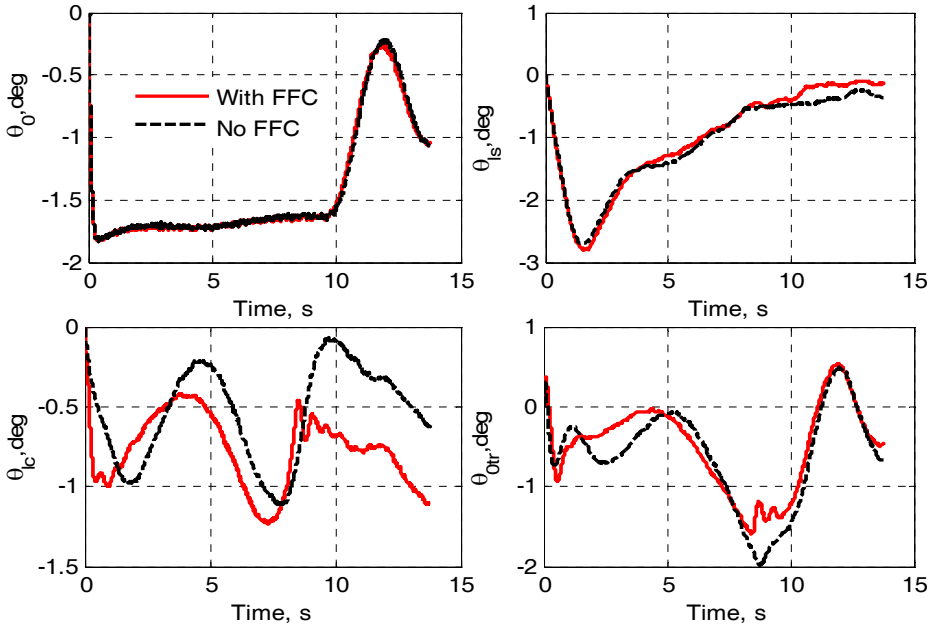

Fig. 9.10 Results showing control efforts from 2DOF with and without FFC for ship landing with disturbance and measurement noise (Lynx-like helicopter)

The results shown in Fig. 9.9 and Fig. 9.10 are quite similar to the ones obtained in Section 8.3 in that the tracking performance with the FFC is better than without the FFC. This is evident in the *p* and q channels and, especially, in the channel $\dot{\Psi}$. In addition, the control efforts with or without the FFC are not very different. The shapes of these control inputs are similar to the results obtained from the inverse simulation for the nonlinear Lynx model, as shown in Fig. 9.3. For instance, the *sinusoidal* shape is present in the first-half time period of the lateral cyclic channel and in the later-half period for the tail rotor channel.

9.3.2 Ship landing based on the linear Lynx helicopter model

The model investigated in this section is linearised from the nonlinear Lynx helicopter model at a forward speed 10 knots. The selection of this equilibrium point is due to the fact that the helicopter has the same initial cruising speed as the ship. In addition, The FBC is also designed by the H_∞ control algorithm, as already illustrated in detail in Chapter 7 (see Fig. 7.1). However, the practical process shows the difficulty of selecting the weighting functions if the selection of the six output channels is equal to ones in Table 7.1. Consequently, to guarantee the good convergence of the H_∞ algorithm, the six output channels are selected to be: $\dot{H}$, u, v, $\dot{\Psi}$, p, and q. This is different from the case in Section 9.3.1.

The weighting functions $\boldsymbol{W}_1(s)$, $\boldsymbol{W}_2(s)$, and $\boldsymbol{W}_3(s)$ selected for this linear model are shown as follows:

$$\begin{aligned}\boldsymbol{W}_1(s) = diag\{&0.6\cdot\frac{s+4}{s+0.001}, 0.48\cdot\frac{s+3.6}{s+0.001}, 0.7\cdot\frac{s+5.6}{s+0.002},\\ &0.6\cdot\frac{s+2}{s+0.002}, 0.02\cdot\frac{s+6.2}{s+0.005}, 0.02\cdot\frac{s}{(s+4)(s+4.5)}\}\end{aligned} \tag{9.7}$$

$$\boldsymbol{W}_2(s) = diag\{0.5\cdot\frac{s+0.001}{s+10}, 0.5\cdot\frac{s+0.001}{s+10}, 0.6\cdot\frac{s+0.001}{s+10}, 0.5\cdot\frac{s+0.001}{s+10}\} \tag{9.8}$$

$$\boldsymbol{W}_3(s) = diag\{1,\quad 1,\quad 1,\quad 1,\quad 0.1,\quad 0.1\} \tag{9.9}$$

The plots of the singular-value of the sensitivity function (***S***) and the complementary sensitivity function (***T***) are shown in the following:

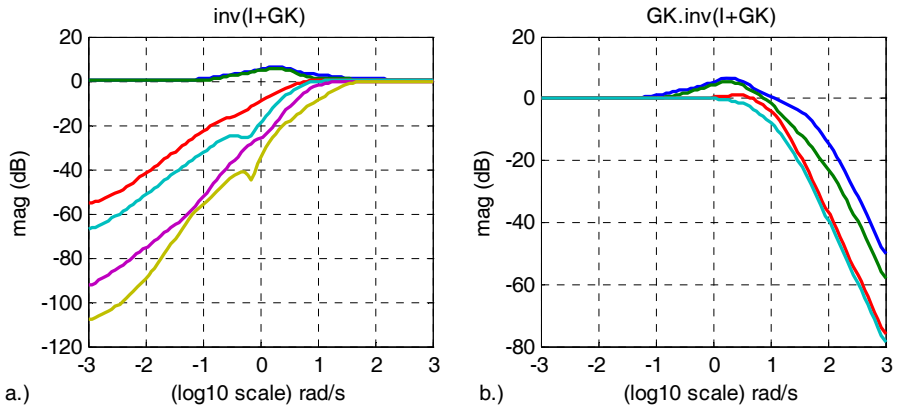

Fig. 9.11 The plot of $(\mathbf{I}+GK)^{-1}$ (a) and $GK(\mathbf{I}+GK)^{-1}$ (b)

The same analysis given in Chapter 7 can be applied to explain that the design specifications have been well achieved in Fig. 9.11. After the design of the FBC, the procedure followed to design the FFC by the inverse simulation method is essentially the same as that in Chapter 3. Therefore, as presented there, the first stage is to find a suitable Δt value. From calculation, the linear system being investigated is found to have two RHP zeros and five zeros located at the origin. By following the methodology introduced in Chapter 3, the magnitude versus the sampling rate is plotted as follows:

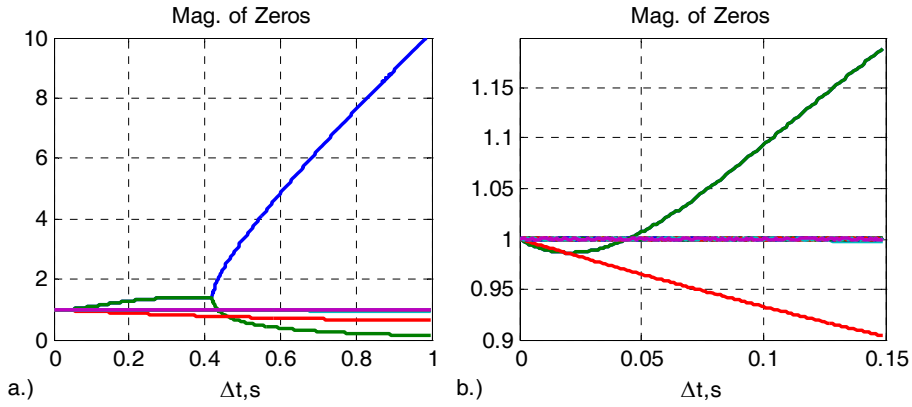

Fig. 9.12 Variation of magnitude of the zeros with Δt

Fig. 9.12(a) shows that there always are some zeroes whose amplitudes are large than one when the value Δt is increased. Fig. 9.12(b) is an enlarged interval [0, 0.15 s] which shows that the amplitudes of all zeros are not larger than one when the Δt value is less than 0.04 s. Therefore Δt = 0.02 s is selected by taking account of the accuracy and convergence speed. The final simulation combines the FBC with the FFC designed through inverse simulation.

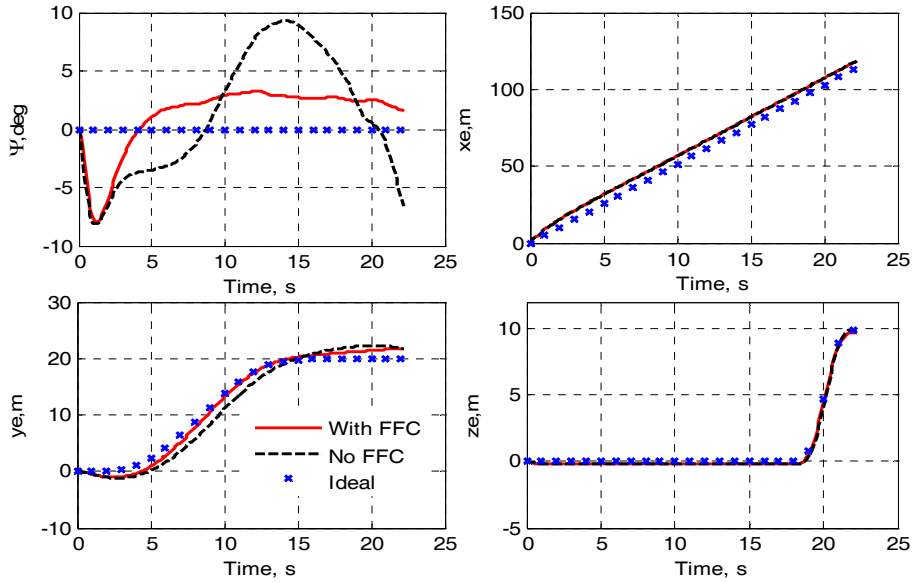

Fig. 9.13 Results from 2DOF with FFC and without FFC for ship landing with disturbance and measurement noise (Lynx helicopter)

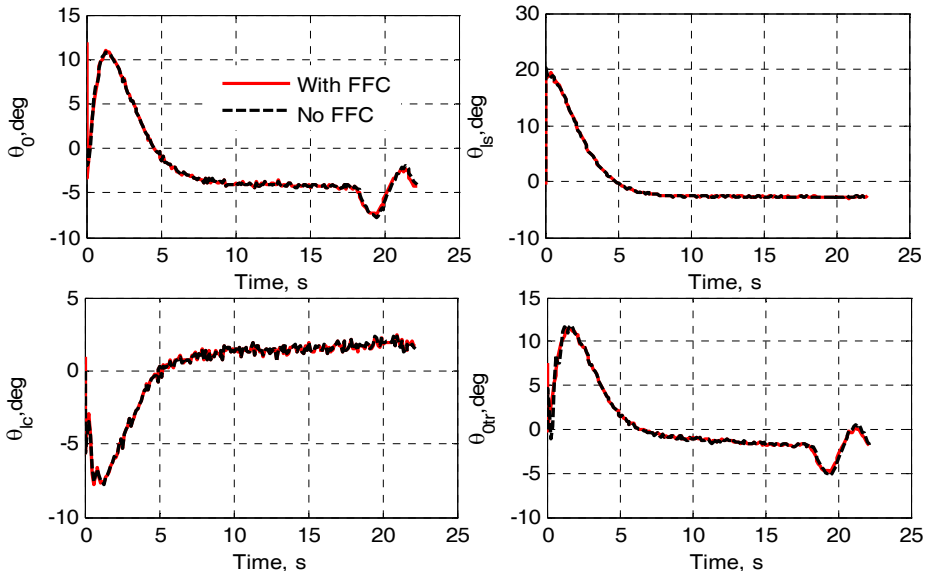

Fig. 9.14 Results showing control efforts from 2DOF with and without FFC for ship landing with disturbance and measurement noise (Lynx helicopter)

The tracking performances achieved for the x_e and z_e channels shown in Fig. 9.13 are not significantly different with and without the inclusion of the FFC. In fact, both channels agree well with the ideal trajectories and thus meet the requirements. However, the y_e channel shows results in which the performance with the FFC is slightly better than for the case without the FFC. This is more obvious in the $\dot{\Psi}$ channel where it seems likely that the results become unbounded with time for the case without the FFC and the outcome with the FFC shows good tracking performance and no signs of tendency of larger divergence.

As far as the control efforts are considered in Fig. 9.14, the collective channel and the tail rotor channel records correspond to those in Fig. 9.3 and can be interpreted with full physical meaning. In addition, for collective channel and lateral channels, it seems that these two channels are affected by minor level noise. However, the noise may not be a problem since the controlled model could be considered as a low-pass filter, as demonstrated by the smooth tracking performance observed in output channels in Fig. 9.13.

This application again shows the benefit of introducing FFC in that it improves the tracking performance. Moreover, because the investigated system is a NMP and nonhyperbolic system, the implementation of the classical model inversion techniques would be quite complicated (Devasia *et al*., 1996, Wang & Chen, 2001). However, inverse simulation can overcome the NMP and nonhyperbolic problem in a more straightforward and causal way.

Chapter 10

Conclusions and Future Work

Contents

This chapter presents the main conclusion of the thesis and highlights the original contributions. In addition, suggestions are made for future work to build on the foundation established here. .

10.1 Conclusions

In the introduction which forms the first chapter it is stated that the main aims and objectives of this thesis. To accomplish these tasks, a series of investigations have been carried out and two new techniques for inverse simulation have been developed and implemented in this thesis. In addition, the idea of designing the FFC through inverse simulation has been realized and validated for a number of practical applications involving ship and aircraft models.

The complexity and tediousness of the traditional model inversion techniques when applied to the type of models encountered in the marine and aerospace fields, especially for NMP systems, has been the motivation for seeking some other approach that is easier to apply. Inverse simulation provides a possible alternative in that it achieves the same objective as model inversion, although using an entirely different methodology. As stated in Chapter 2,

model inversion based methods obtain the input through inversion of a nonlinear dynamic system in advance whereas inverse simulation does this through a numerical process.

Chapter 3 describes a comprehensive investigation of the relationship between inverse simulation and model inversion. The link between the two approaches is presented by dividing the most widely used inverse simulation process into two stages – the discretization process and the iterative procedures based upon the NR algorithm. By a suitable discretization interval and the guaranteed stability of the NR algorithm, for the case of MP systems, inverse simulation is shown to provide a viable alternative approach. This is easier to apply and more feasible in terms of practical implementation, compared with traditional model inversion techniques. Moreover, the work carried out shows that for the case of NMP systems the discretization process can contribute significantly to the successful application of the inverse simulation approach through zero redistribution. This is different from the method of Yip and Leng (1998) since it removes the assumption of a constant Jacobian matrix and fast convergence to achieve the approximation of the NR algorithm. Moreover, the analysis presented in this thesis show that the discretization process and the NR algorithm play significant roles of determining the stability of the whole inverse simulation process.

The ideas relating to inverse simulation methods have been validated and illustrated through applications involving a nonlinear HS125 fixed-wing aircraft model, a linear SISO NMP system, and an eighth-order linear Lynx-like helicopter model. The results from these applications prove the effectiveness of inverse simulation over model inversion. In addition, all the cases considered show that the computational overheads of the proposed approach based on inverse simulation are modest, regardless of whether nonlinear or linear systems are being considered. Moreover, the speed may be further increased by increasing the sampling interval without adversely affecting the accuracy of the results. All these results provide support for the innovative idea presented in this thesis that it may be possible and practical for inverse simulation to replace model inversion in the output-tracking field or other corresponding domains.

Chapter 4 reviews and analyses three issues relating to traditional inverse simulation algorithms. These involve the high-frequency oscillation phenomenon, the redundancy problem, and the phenomenon of constraint oscillations, as outlined in Chapter 1. The first two problems are addressed by reviewing historical contributions and most effort has been devoted to investigation of the third issue. The findings contrast with traditional explanations of the root cause of constraint oscillations. Results from investigations involving the nonlinear Lynx helicopter model show that factors such as the sampling rate, the type of manoeuvres, and the internal dynamics of the model itself all contribute to this special phenomenon. In addition, analysis involving the linearised helicopter model around specific trim points has also been useful in this investigation. The reasons of influence from the sampling rate are based on the increased information contained in the outputs as a result of using smaller sampling rate values. The effects associated with the severity of the manoeuvre or the input-output analytic relationship that leads to more distinct oscillations are believed to result from the highly coupled nature of helicopter dynamics and the relatively complex form of helicopter model being investigated. Finally, the internal dynamics is believed to have a critically important role in the generation of constraint oscillations. There is much evidence of consistency, in terms of frequency and the general nature of observed oscillations, between the overall dynamic characteristics of an inverse simulation model and characteristics of the underlying forward model in terms of the zeros (of a SISO linearised description) or the internal dynamics (in the more general case such as a multivariable nonlinear model).

In Chapter 5 the new SA-based procedure for inverse simulation is developed and validated. The new methodology shows advantages over the traditional inverse simulation approaches in the number of respects. Firstly, it can deal with the redundancy problem in a natural way. Secondly, the development process of this new technique allows for calculation of the Jacobian matrix by solving a sensitivity equation, at the cost of computational complexity. Therefore, this avoids the traditional numerical approximations to calculate the Jacobian matrix and thereby provides greater accuracy in terms of the results.

The results from a nonlinear HS125 aircraft model and a nonlinear Lynx helicopter model for different manoeuvres such as the hurdle-hop and pop-up manoeuvres show that this new SA-based technique is a reliable and flexible tool for inverse simulation. Furthermore, it also shows increased stability, better convergence properties, and higher accuracy in comparison with the traditional approaches. The high-frequency oscillations that appear in the traditional inverse simulation process are largely eliminated in the SA approach by increasing the integration number. The only disadvantage of the SA approach is an increase of the computation time but this is believed to be acceptable in practice.

Chapter 6 describes work that provides insight into problems of inverse simulation associated with input saturation and discontinuous manoeuvres, which have traditionally been ignored in the inverse simulation field. A new derivative-free procedure for inverse simulation, based on the constrained NM algorithm, is proposed to overcome these problems. This proposed approach adopts one-step forward-simulation input transformations of the integration-based structure before applying a pattern-search form of optimisation method. Therefore, it can avoid use of the augmented Lagrangian method to deal with the constrained conditions.

A number of cases have been studied using five nonlinear marine models which represent three different situations with respect to the different numbers of inputs and outputs. Results from three manoeuvres have been obtained – the turning-circle manoeuvre, a zigzag type of manoeuvre, and a pullout manoeuvre – and these prove the effectiveness of the new method in terms of improved convergence and numerical stability for cases involving input saturation or discontinuous manoeuvres. This improvement in performance compared with traditional inverse simulation algorithms also provides a good chance to understand better some of the well known numerical problems that commonly occur. In addition, the results also show a multi-solution phenomenon in the case of severe manoeuvres such as a zigzag and for complex models such as an AUV.

Chapter 7 develops FBCs using the mixed-sensitivity H_∞ optimisation method for the Norrbin ship model and the linear Lynx-like helicopter model. The designed FBCs show

perfect tracking performance against external disturbances and measurement noise for both the models investigated. In particular, for the case of the Norrbin ship model the control system shows significant robustness in that the same FBC works well for large changes of the forward speed. However, the FBC designed for the linear Lynx-like helicopter model fails in the case of more severe (but more artificial) manoeuvres.

Subsequently, in addition to the mixed-sensitivity H_∞ optimisation method, the LQ method is also implemented for the Son and Nomoto Container ship model. The simulation results show that the both types of controller considered provide good robustness to changes of forward speed within the range 5 m/s, 7.3 m/s, 10 m/s, and 13 m/s. However, the tracking performance will deteriorate significantly in the presence of measurement noise. Moreover, the required control efforts tends to become unacceptable large. In contrast, none of these deficiencies has been found in the results with the mixed-sensitivity H_∞ optimisation method and therefore no further design efforts are required to deal with measurement noise. Furthermore, good rudder-roll reduction is achieved by the H_∞ design with similar rudder control effort while maintaining good steering course.

Chapter 8 has achieved two main goals. One is that this chapter presents a systematic analysis of the influence of the uncertainties within the linear controlled model on the performance of the FFC in the 2DOF control scheme. The other goal is the successful development and implementation of the FFC based on inverse simulation combined with an effective H_∞ controller based on the FBC developed in Chapter 7. This is one of the main contributions made in this thesis.

The validity and effectiveness of the approach are demonstrated by two applications. The first case study involves an eighth-order linearised NMP and hyperbolic helicopter model involving four groups of manoeuvres with differing levels of severity. The causal and bounded results prove the feasibility and flexibility of inverse simulation for replacement of more complex model-inversion techniques in such situations. The inverting procedure would be complicated for an application such as this if the traditional model inversion techniques were adopted. For the second case study, the linear and nonlinear robust

feedforward tracking controllers, based on the constrained NM method introduced in Chapter 6, have been successfully implemented for the traditional 2DOF control structure for the nonlinear Container ship model. The results prove the effectiveness of this form of controller in the context of ship steering and roll stabilization as well as the ability to overcome the effects of measurement noise and providing good disturbance rejection. In addition, the nonlinear FFC shows better control performance than the linear FFC because more information is contained in the feedforward channel.

Chapter 9 has extended the application of inverse simulation to the helicopter ship landing field. At first, two cases, with and without atmospheric disturbances, have been investigated during the ship landing process. The results from the inverse simulation process with the wind disturbance are similar in form to those found in still air conditions, just offset in terms of velocity. This may suggests that the control strategies for landing on a deck might be similar for the situations involving headwind and tailwind. By following the design procedures discussed in Chapter 8, inverse simulation is used to design the FFCs for the Lynx-like helicopter model and the linear Lynx helicopter model. The results from these investigations again show improved tracking performance, as well as demonstrating yet again the effectiveness and feasibility of inverse simulation to design the FFC instead of using traditional model inversion.

10.2 Future work

Some suggestions and comments relating to possible future work have already been mentioned in the main chapters. The most significant areas for further research are listed below:

i. Within this thesis techniques are described which allow inverse simulation techniques to be applied to nonlinear MP and linear NMP systems. The results from a series of investigation have shown the effectiveness of these methods for these two classes of systems. However, the investigation of inverse simulation for the case of nonlinear NMP systems requires further consideration and effort. Although inverse simulation

has been applied successfully to nonlinear NMP systems such as the nonlinear Lynx helicopter model and the nonlinear Container ship model, the reasons why good results are achieved in these applications are still not fully understood. Even for nonlinear MP systems, further research is needed because such systems may be transformed into NMP systems as the operating conditions change during the tracking of an ideal trajectory.

ii. In this thesis, inverse simulation has been used to achieve casual inversion in a feasible and straightforward way for NMP systems. This is one way in which inverse simulation shows an advantage over model inversion in solving inverse problems. Compared with the complexity of model inversion techniques, this causal process involving inverse simulation may allow the application of this inverse-model based control approach to real-time situations. This appears, given the current capabilities of modern processors, particularly viable in the ship control field, due to the relatively slow dynamics of typical vessels. Because of the limited research time available and the objectives of the project, issues associated with practical real-time applications based on inverse simulation have not been investigated. These applications, as well as further performance comparisons involving model inversion techniques and inverse simulation applied to the same systems, are important in future research plans. It may appear that further research could lead to more effective and efficient implementation of the proposed control algorithm in real-time.

iii. For applications involving the marine system models, a multi-solution phenomenon appears in the results from the inverse simulation process based on the constrained NM approach, as shown in Chapter 6. This multi-solution phenomenon may have potential advantages in dealing with control reallocation and might allow the optimal control effort to be found by modification of the cost-function definition. This is an interesting field that might be worthy of future research.

iv. An anonymous reviewer of one of the published papers has raised an interesting point that the state derivatives in the following equation [Eq. (2.33)]

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$$

could be experimentally measured, instead of being computed through the equation below [Eq. (2.40)].

$$\boldsymbol{x}(t_{k+1}) = \int_{t_k}^{t_{k+1}} \dot{\boldsymbol{x}}(t)dt + \boldsymbol{x}(t_k)$$

If this were done it might provide another potential advantage for the use of inverse simulation compared with model inversion for control system applications. In the inverse simulation process sensor readings (e.g. accelerometer outputs) could be integrated directly to provide FFC terms, for systems for which no mathematical model can readily be established. This could be a possible area of further investigation.

v. In Chapter 8, the influence of the uncertainties in the control objectives on the performance of the FFC has been assessed. However, all those discussions are based on linearised models. In the previous work, only a qualitative statement is available about the possible improvement in tracking performance of the control system with the nonlinear FFC compared with linear FFC. This improvement arises as a result of the fact that more information is included in the feedforward channel in the nonlinear case. Therefore, further efforts are required to investigate the influence of uncertainties on the feedforward channel for a nonlinear system.

vi. Chapter 9 has successfully demonstrated a ship landing control problem with the FFC for the linear Lynx helicopter model. However, the research has not yet been applied to the nonlinear model. This is due to time constraints as well as the complexity of the nonlinear helicopter model. However, it is a further interesting area for future research.

References

Aguiar, A.P, Hespanha, J.P., & Kokotovic, P.V. 2005. Path-following for nonminmum phase systems removes performance limitations. *IEEE Transactions on Automatic Control*, **50**(2), 234-9.

Al-Hiddabi, S.A. & McClamroch, N.H., 2002. Tracking and maneuver regulation control for nonlinear nonminimum phase systems: application to flight control. *IEEE Transactions on Control Systems Technology*, **10**(6), 780-792.

Alfaro-Cid, E., McGookin, E.W., Murray-Smith, D.J., & Fossen, T.I. 2005. Genetic algorithms optimisation of decoupled sliding mode controllers: simulated and real results. *Control Engineering Practice*, **13**(6), 739-748.

Anderson, D. 2003, Modification of a generalised inverse simulation technique for rotorcraft flight. *Proceedings of the Institution of Mechanical Engineers*, *Part G: Journal of Aerospace Engineering*, **217**(2), 61-73.

Angeli, D., Chitour, Y., & Marconi, L. 2005. Robust stabilization via saturated feedback. *IEEE Transactions on Automatic Control*, **50**(12), 1997-2014.

Anon, 2000. *Aeronautical design standard performance specification handling qualities requirements for military rotorcraft ADS-33E-PRF*. US Army Aviation and Missile Command, Aviation Engineering Directorate, Redstone Arsenal, Alabama.

Anon, 2007. *Flight Direction Control*. www.thaitechnics.com /helicopter / tg5/control.jpg. Thai Technics. Com.

Avanzini, G. 2004. Two time-scale approach to inverse simulation: algorithm development and application. Department of Aeronautical and Space Engineering, Politecnico di Torino, Italy. Private Communication (presentation materials).

Avanzini, G. & de Matteis, G. 2001. Two-timescale inverse simulation of a helicopter model. *Journal of Guidance, Control, and Dynamics*, **24**(2), 330-9.

Avanzini, G., de Matteis, G., & de Socio, L.M. 1998. Natural description of aircraft motion. *Journal of Guidance, Control, and Dynamics*, **21**(2), 229-233.

Avanzini, G., de Matteis, G., & de Socio, L.M. 1999. Two-timescale-integration method for inverse simulation. *Journal of Guidance, Control, and Dynamics*, **22**(3), 395-401.

Bagiev, M. 2006. Rotorcraft inverse simulation package [Ver. 2.1]. Dept of Aerospace Engineering, University of Glasgow, UK.

Boskovic, J.D. & Mehra, R.K. 2002. Control allocation in overactuated aircraft under position and rate limiting. *Proceedings of the 2002 American Control Conference*; May 8-10, Anchorage, AK, US, 791-6.

Boyle, D.P. & Chamitoff, G.E. 1999. Autonomous maneuver tracking for self-piloted vehicles. *Journal of Guidance, Control, and Dynamics*, **22**(1), 58-67.

Braatz, R.D., Rusli, E., Drews, T.O., Ma, D.L., & Alkire, R.C, 2006. Robust nonlinear feedback-feedforward control of a coupled kinetic Monte Carlo finite difference simulation. *Journal of Process Control*, **16**(4), 409-17.

Bradley, R. & Thomson, D.G. 1993. The development and potential of inverse simulation for the quantitative assessment of helicopter handling qualities, piloting vertical flight aircraft. *Proceedings of the AHS/NASA Conference 'Piloting Vertical Flight Aircraft: Flying Qualities and Human Factors'*; Jan, San Francisco, US, 359-371

Brockett, R.W. 1965. Poles, zeros, and feedback: state space interpretation. *IEEE Transactions on Automatic Control*, **10**(2), 129-135.

Burdisso, R.A. & Fuller, C.R. 1994. Feedforward controller design by eigenvalue assignment. *Journal of Guidance, Control, and Dynamics*, **17**(3), 466-472.

Byrns, E.V., Jr. & Calise, A.J. 1994. Approximate recovery of H infinity loop shapes using fixed-order dynamic compensation. *Journal of Guidance, Control, and Dynamics*, **17**(3), 458-465.

Celi, R. 2000. Optimization-based inverse simulation of a helicopter slalom manoeuvre. *Journal of Guidance, Control, and Dynamics*, **23**(2), 289-297.

Che, J. & Chen, D. 2001. Automatic landing control using H infinity control and stable inversion. *Proceedings of the IEEE Conference on Decision and Control*; Dec 4, Orlando, FL, US, 241-246.

Chelouah, R. & Siarry, P. 2003. Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. *European Journal of Operational Research*, **148**(2), 335-348.

Cheney, W. & Kincaid, D. 2004, *Numerical mathematics and computing*, 5th ed. America: Brooks/Cole Publishing Company.

de Matteis, G. de Socio, L.M., & Leonessa, A., 1995. Solution of aircraft inverse problems by local optimization. *Journal of Guidance, Control, and Dynamics*, **18**(3), 567-571.

Devasia, S. 1997. Output tracking with nonhyperbolic and near nonhyperbolic internal dynamics: helicopter hover control. *Journal of Guidance, Control, and Dynamics*, **20**(3), 573-580.

Devasia, S. 1999. Approximated stable inversion for nonlinear system with nonhyperbolic internal dynamics. *IEEE Transactions on Automatic Control*, **44**(7), 1419-1425.

Devasia, S. 2000. Robust inversion-based feedforward controllers for output tracking under plant uncertainty. *Proceedings of the American Control Conference*; Jun 28-30, Chicago, Illinois, US, 497-502.

Devasia, S. 2002. Should model-based inverse inputs be used as feedforward under plant uncertainty?. *IEEE Transactions on Automatic Control*, **47**(11), 1865-1871.

Devasia, S., Chen, D., & Paden, B. 1996. Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, **41**(7), 930-942.

Devasia, S. & Paden, B. 1998. Stable inversion for nonlinear nonminimum-phase time-varying system. *IEEE Transactions on Automatic Control*, **43**(2), 283-8.

Do, K.D. & Pan, J. 2006. Underactuated ships follow smooth paths with integral actions and without velocity measurements for feedback: theory and experiments. *IEEE Transactions on Control Systems Technology*, **14**(2), 308-322.

Donha, D.C., Desanj, D.S., Katebi, M.R., & Grimble, M.J. 1998. H infinity adaptive controllers for auto-pilot applications. *International Journal of Adaptive Control and Signal Processing*, **12**(8), 623-648.

Dorato, P. 1969. On the inverse of linear dynamical systems. *IEEE Transactions on Systems Science and Cybernetics*, **5**(1), 43-48.

Duda, R.O., Hart, P.E., & Stork, D.G. 2000. *Pattern classification,* 2nd ed. New York: John Wiley & Sons, Inc.

Errico, J.D. 2005. *Bound constrained optimization*. Computer Program. MATLAB and Simulink Centre, the MathWorks, Inc.

Escande, B. 1997. Nonlinear dynamic inversion and LQ techniques. In Magni, J.F., Bennani, S. & Terlouw, J. (eds.). *Robust Flight Control: A Design Challenge*. London: Springer Verlag, 523-540.

Fales, R. & Kelkar, A. 2005. Robust control design for a wheel loader using mixed sensitivity H-infinity and feedback linearization based methods. *Proceedings of the 2005 American Control Conference*; Jun 8-10, Portland, OR, US, 4381-6.

Fancis, B.A. & Wonham, W.M. 1976. The internal model principle of control theory. *Automatica*, **12**(5), 457-465.

Fang, M.C. & Luo, J.H. 2005. The nonlinear hydrodynamic model for simulating a ship steering in waves with autopilot system. *Ocean Engineering*, **32**(11-12), 1486-1502.

Fossen, T. I. 1994. G*uidance and control of ocean vehicles*. UK: John Wiley & Sons Ltd.

Fossen, T. I., Perez T., Smogeli Q.N., & Sorensen A.J. 2005. *Guidance, navigation and control toolbox*. Norwegian University of Science and Technology, Trondheim.

Gao, C. & Hess, R.A. 1993. Inverse simulation of large-amplitude aircraft manoeuvres. *Journal of Guidance, Control, and Dynamics*, **16**(4), 733-7.

Giusto, A. & Paganini, F. 1999. Robust synthesis of feedforward compensators. *IEEE Transactions on Automatic Control*, **44(**8), 1578-1582.

Graichen, K., Hagenmeyer, V., & Zeitz, M. 2005. A new approach to inversion-based feedforward control design for nonlinear systems. *Automatica*, **41**(12), 2033-2041.

Gray, G.J. & Grünhagen, W.V. 1998. An investigation of open loop and inverse simulation as nonlinear model validation tools for helicopter flight mechanics. *Mathematical and Computer Modelling of Dynamical Systems*, **4**(1), 32-57.

Haverkort, B.R. & Meeuwissen, A.M.H. 1995. Sensitivity & uncertainty analysis of markov-reward models. *IEEE Transactions on Reliability*, **44**(1), 147-154.

Hess, R.A., Gao, C., & Wang, S.H. 1991. A generalized technique for inverse simulation applied to aircraft maneuvers. *Journal of Guidance, Control, and Dynamics*, **14(**5), 920-6.

Hirschorn, R.M. 1979. Invertibility of multivariable nonlinear control system. *Journal of Guidance, Control, and Dynamics*, **AC-24**(6), 855-865.

Houston, S.S. 1998. Identification of autogyro longitudinal stability and control characteristics. *Journal of Guidance, Control, and Dynamics*, **21**(3), 391-9.

Hu, S.S., Yang, P.H., Juang, J.Y., & Chang, B.C. 2003. Robust nonlinear ship course-keeping control by H I/O linearization and synthesis. *International Journal of Robust and Nonlinear Control*, **13**(1), 55-70.

Huang, B., Shah, S.L., & Miller, R. 2000. Feedforward plus feedback controller performance assessment of mimo systems. *IEEE Transactions on Control Systems Technology*, **8**(3), 580-7.

Hunt, L. R. & Meyer, G. 1997. Stable inversion for nonlinear system. *Automatica*, **33**(8), 1549-1554.

Isidori, A. 1989. *Nonlinear control systems: an introduction*, **2**nd ed. London: Springer.

Isidori, A. & Byrnes, C.I. 1990. Output regulation of nonlinear systems. *IEEE Transactions on Automatic Control*, **35**(2), 3544-8.

Jones, R. 1936. A s*implified application of the method of operators to the calculation of disturbed motions of an airplane*, *N*ACA TR560.

Kato, O. & Saguira, I. 1986. An interpretation of airplane general motion and control as inverse problem. *Journal of Guidance, Control, and Dynamics*, **9**(2), 198-204.

La Civita, M., Papageorgiou, G., Messner, W.C., & Kanache, T. 2003. Integrated modeling and robust control for full-envelope flight of robotic helicopters, *Proceedings of IEEE International Conference on Robotics and Automation*; Sep 14-19, Taipei, Taiwan, 552-7.

Lagarias, J.C., Reeds, J.A., Wright, M.H., & Wright, P.E. 1998. Convergence properties of the nelder-mead simplex method in low dimensions. *Siam. J. Optim*., **9**(1), 112-147.

Lan, W., Chen, B. M., & He, Y. 2006. On improvement of transient performance in tracking control for a class of nonlinear systems with input saturation. *Systems and Control Letters*, **55**(2), 132-8.

Lee, S. & Kim, Y. 1997. Time-domain finite element method for inverse problem of aircraft maneuvers. *Journal of Guidance, Control, and Dynamics*, **20**(1), 97-103.

Lewis, R.M., Torczon, V., & Trosset, M.W. 2000. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, **124**(1-2), 191-207.

Li, S. 1997. Nonlinear H infinity controller design for a class of nonlinear control systems. *Proceedings of the IECON'97 23rd International Conference on Industrial Electronics, Control, and Instrumentation*; Nov 9-14, New Orleans, La, US, 291-294.

Lim, S. & Chan, K. 2003. Coordinated feedforward and feedback control for fast repositioning of uncertain flexible systems. *Proceedings of the American Control Conference*; Jun 4-6, Denver, CO, US, 4799-4804.

Limebeer, D.J.N., Kasenally, E.M., & Perkins, J.D. 1993, On the design of robust two degree of freedom controllers. *Automatica*, **29**(1), 157-168.

Lin, K. 1993. Comment on generalized technique for inverse simulation applied to aircraft maneuvers. *Journal of Guidance, Control, and Dynamics*, **16**(6**)**, 1196-7.

Loo, M., McGookin, E. W., & Murray-Smith, D. J. 2005. Application of inverse model control to IFAC benchmark models. 4th International Conference on Advanced Engineering Design (AED 2004); Sep 5-8, Glasgow, UK.

Lóez E, Velasco F. J., Moyano E., & Rueda y T. M. 2004. Full-scale manoeuvering trials simulation. *Journal of Maritime Research*, **1**(3), 37-50.

Lu L, Murray-Smith DJ, and McGookin EW (2006a). Applications of inverse simulation within the model-following control structure. *Proc. of International Control Conference*; ICC2006, Aug. 30th – Sept 1st, Glasgow, UK.

Lu L, Murray-Smith DJ, and McGookin EW (2006b). Relationships between model inversion and inverse simulation techniques. *Proc. of 5th MATHMOD*; Feb 8-10, Vienna, Austria.

Lu L, Murray-Smith DJ, and McGookin EW (2007a). Feedforward controller design from a constrained derivative-free inverse simulation process. *Control Engineering Practice*. Submitted for Publication.

Lu L, Murray-Smith DJ, and McGookin EW (2007b). Investigation of inverse simulation for design of feedforward controllers. *Journal of Mathematical and Computer Modelling of Dynamical Systems*. Accepted for Publication.

Lu L, Murray-Smith DJ, and Thomson DG (2007c). A sensitivity-analysis method for inverse simulation. *Journal of Guidance, Control, and* Dynamics, **30**(1), 114-121.

Lu L, Murray-Smith DJ, and Thomson DG (2007d). Issues of numerical accuracy and stability in inverse simulation. *Simulation Modelling Practice and Theory*. Under Revision.

Luersen, M.A., Le Richem, R., & Guyon, F. 2004. A constrained, globalised, and bounded nelder-mead method for engineering optimization. *Structural and Multidisciplinary Optimization,* **27**(1-2), 43-54.

Luo, C.C., Liu, R.F., Yang, C.D., & Chang, Y.H. 2003. Helicopter H infinity control design with robust flying quality. *Aerospace Science and Technology,* **7**(2), 159-169.

McGeoch, D.J. 2005. Helicopter flight control system design using sliding mode theory: application to handling qualities and shipboard landing. PhD Thesis. Department of Electrical & Electronic Engineering, University of Glasgow.

Mammar, S., Koenig, D., & Nouveliere, L. 2001. Combination of feedforward and feedback H infinity control for speed scheduled vehicle automatic steering. *Proceedings of the American Control Conference*; Jun 25-27, Arlington, VA, 684-9.

McGookin, E.W., Murray-Smith, D.J., Li, Y., & Fossen, T.I. 2000. Ship steering control system optimisation using genetic algorithms. *Control Engineering Practice*, **8**(4), 429-443.

Mickle, M.C., Huang, R., & Zhu, J.J. 2004. Unstable, nonminimum phase, nonlinear tracking by trajectory linearization control. *Proceedings of the IEEE Conference on Decision and Control*; Sep 2-4, Taipei, Taiwan, 812-8.

Min, K.-W., Chung, L., Joo, S.-J., & Kim, J. 2005. Design of frequency-dependent weighting functions for H2 control of seismic-excited structures. *Journal of Vibration and Control*, **11**(1), 137-157.

Moghaddam, M. M. & Moosavi, S. F. 2005. Robust maneuvering control design of an aircraft via dynamic inversion and μ-synthesis. *Proceedings of the I MECH E Part G Journal of Aerospace Engineering*, **219**(1), 11-8.

Muramatsu, E. & Watanabe, K. 2004. Two-degree-of-freedom control with adaptive inverse model. *Proceedings of the SICE Annual Conference*; Aug 4-6, Sapporo, Japan, 1677-1680.

Murray-Smith, D.J. 2000. The inverse simulation approach: a focused review of methods and applications. *Mathematics and Computers in Simulation*, **53**(4-6), 239-247.

Nelder, J.A. & Mead, R. 1965. A simplex method for function minimization. *Computer Journal*, **7**, 308-313.

Ortega, M.G. & Rubio, F.R. 2004. Systematic design of weighting matrices for the H-infinity mixed sensitivity problem. *Journal of Process Control*, **14**(1), 89-98.

Osborne, R.C., Adams, R.J., Hsu, C.S., & Banda, S.S. 1994. Reduced-order H infinity compensator design for an aircraft control problem. *Journal of Guidance, Control, and Dynamics*, **17**(2), 341-5.

Postlethwaite, I., Prempain, E., Turkoglu, E., Turner, M.C., Ellis, K., & Gubbels, A.W. 2005. Design and flight testing of various H infinity controllers for the bell 205 helicopter. *Control Engineering Practice*, **13**(3), 383-398.

Qui, L. & Davison, E.J. 1993. Performance limitations of nonminimum phase systems in the servomechanism problem. *Automatica*, **29**(2), 337-349.

Ramakrishna, V., Hunt, L.R., & Meyer, G. 2001. Parameter variations, relative degree, and stable inversion. *Automatica*, **37**(6), 871-880.

Ratto, M. 2001. Sensitivity analysis in model calibration: GSA-GLUE approach. *Computer Physics Communications*, **136**(3), 212-224.

Reiner, J., Balas, G.J., & Garrard, W.L. 1995. Robust dynamic inversion for control of highly maneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, **18**(1), 18-24.

Rosenwasser, E. & Usopov, R. 2000. *Sensitivity of automatic control systems*. London: CRC Press.

Rutherford, S. & Thomson, D.G. 1996. Improved methodology for inverse simulation. *Aeronautical Journal*, **100**(993), 79-86.

Rutherford, S. & Thomson, D.G. 1997. Helicopter inverse simulation incorporating an individual blade rotor model. *Journal of Aircraft*, **34**(5), 627-634.

Sain, M.K. & Massey, J.L. 1969. Invertibility of linear time-invariant dynamical systems. *IEEE Transactions on Automatic Control*, **AC-14**(2), 141-9.

Saltelli, A. & Scott, M. 1997. Guest editorial: the role of sensitivity analysis in the corroboration of models and its link to model structural and parametric uncertainty. *Reliability Engineering and System Safety*, **57**(1), 1-4.

Sastry, S. 1999. *Nonlinear systems: analysis, stability, and control*. New York: Spring-Verlag.

Sato, J., Ueda, T., & Ohmori, H. 2004. Sensitivity analysis and identification of time varying parameters in wastewater treatment system based on activated sludge models, *Modelling and Control For Participatory Planning and Managing Water Systems*; Sept 29th – Oct. 1st, Venice, Italy.

Sentoh, E. & Bryson, A. 1992. Inverse and optimal control for desired outputs. *Journal of Guidance, Control, and Dynamics*, **15**(3), 687-691.

Silverman, L.M. 1969. Inversion of multivariable linear systems. *IEEE Transactions on Automatic Control*, **AC-14**(3), 270-6.

Skogestad, S. & Postlethwaite, I. 1996. *Multivariable feedback control: analysis and design*, 1st ed. Chichester: JOHN WILEY & SONS,

Soroush, M., Valluri, S., & Mehranbod, N. 2005. Nonlinear control of input-constrained systems. *Computers & Chemical Engineering*, **30**(1), 158-181.

Takahashi, M.D. 1994. H infinity helicopter flight control law design with and without rotor state feedback. *Journal of Guidance, Control, and Dynamics*, **17**(6), 1245-1251.

Tanner, O. & Geering, H.P. 2003. Two-degree-of-freedom robust controller for an autonomous helicopter. *Proceedings of the American Control Conference*; Jun 4-6, Denver, CO, US, 993-8.

Thomson, D.G. 1987. *Evaluation of helicopter agility through inverse solution of the equation of motion*. PhD Thesis, Faculty of Engineering, University of Glasgow, Scotland, UK.

Thomson, D.G. 2004. Mathematical Mmodelling and Simulation of Fixed Wing Aircraft. Faculty of Engineering, University of Glasgow, Scotland, UK.

Thomson, D.G. & Bradley, R. 1990a. Prediction of the dynamic characteristics of helicopters in constrained flight. *Aeronautical Journal*, **94**, 344-354.

Thomson, D.G. & Bradley, R. 1990b. The use of inverse simulation for conceptual design. *16th European Rotorcraft Forum*; Sept, Glasgow, Scotland, UK.

Thomson, D.G. & Bradley, R. 1994. The contribution of inverse simulation to the assessment of helicopter handling qualities. *Proceedings of the 19th ICAS Conference*; Sept 18-23, Anahiem, US, 229-238.

Thomson, D.G. & Bradley, R. 1997. The use of inverse simulation for preliminary assessment of helicopter handling qualities. *Aeronautical Journal*, **101**(1007), 287-294.

Thomson, D.G. & Bradley, R. 1998. The principles and practical application of helicopter inverse simulation. *Simulation Practice and Theory*, **6**(1), 47-70.

Thomson, D.G., Coton, F., & Galbraith, R. 2005. A simulation study of helicopter ship landing procedures incorporating measured flow-field data. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, **219**(5), 411-427.

Thomson, D.G., Taylor, C.D., Talbot, R., Ablett, R., & Bradley, R. 1995. An investigation of piloting strategies for engine failure during takeoff from offshore platforms. *Aeronautical Journal*, **99**(981), 15-25.

Tomovic R. 1963. *Sensitivity analysis of dynamic systems*. New York: McGraw Hill.

Tzeng, C.Y., Goodwin, G.C., & Crisafulli, S. 1999. Feedback linearization design of a ship steering autopilot with saturating and slew rate limiting actuator. *International Journal of Adaptive Control and Signal Processing*, **3**(1), 23-30.

Unar, M.A. 1999. *Ship steering control using feedforward neural networks.* PhD Thesis, Department of Electronics and Electrical Engineering, University of Glasgow, UK.

Visioli, A. 2004. A new design for a PID plus feedforward controller. *Journal of Process Control,* **14**(4), 457-463.

von Grunhagen, W.V., Bouwer, G., Pausder, H.J., Henschel, F., & Kaletka, J. 1996. A high bandwidth control system for the helicopter in-flight simulator ATTHes modelling, performance and applications. In M.B. Tischler, (ed.), *Advances in Aircraft Flight Control*. London: Taylor & Francis, 73-101.

Walker, D.J. & Postlethwaite, I. 1996. Advanced helicopter flight control using two-degree-of-freedom H infinity optimization. *Journal of Guidance, Control, and Dynamics*, **19**(2), 461-8.

Wang, X. & Chen, D. 2001. Causal inversion of nonminimum phase system. *Proceedings of the 40th IEEE Conference on Decision and Control*; Dec 4, Orlando, FL, US, 73-8.

Wang, X. & Chen, D. 2002a. Tip trajectory tracking for a one-link flexible manipulator using causal inversion. *Proceedings of the IEEE Conference on Control Applications*; Sep 18-20, Glasgow, UK, 507-512.

Wang, X. & Chen, D. 2002b. Output tracking control of nonminimum phase systems via causal inversion. *Proceeding of the 45th Midwest Symposium on Circuits and Systems*; Aug 4-7, Tulsa, OK, US, 125-8.

Wang, X. & Ren, Y. 2004. H infinity control of ship steering. *2004 IEEE Conference on Robotics, Automation and Mechatronics*; Dec 1-3, Singapore, 1198-1202.

Wik, T., Fransson, C.M., & Lennartsson, B. 2003. Feedforward feedback controller design for uncertain systems. *Proceedings of the 42nd IEEE Conference on Decision and Control*; Dec 9-12, Maul, Hawaii, US, 5328-5334.

Williams, P. 2005. Aircraft trajectory planning for terrain following incorporating actuator constraints. *Journal of Aircraft*, **42**(5), 1358-1361.

Wolff, S. 2004. A local and globalized, constrained and simple bounded Nelder-Mead method [Ver. 2.0]. Computer Program. Bauhaus University Weimar, Germany.

Yang, C.D., Ju, H.S., & Liu, S.W. 1994. Experimental design of H infinity weighting functions for flight control systems. *Journal of Guidance, Control, and Dynamics*, **17**(3), 544-552.

Yang, C.D., Liu, W.H., & Kung, C.C. 2002. Nonlinear H infinity decoupling control for hovering helicopter. *Proceedings of the 2002 American Control Conferen*ce; May 8-10, Anchorage, AK, 4353-8.

Yang, C.D., Luo, C.C., Liu, S.J., & Chang, Y.H. 2005. Applications of genetic Taguchi algorithm in flight control designs. *Journal of Aerospace Engineering,* **18**(4), 232-241.

Yang, C.D., Tai, H.C., & Lee, C.C. 1994. Systematic approach to selecting H infinity weighting functions for dc servos. *Proceedings of the 33rd IEEE Conference on Decision and Control*; Dec 14-16, Lake Buena Vista, FL, US, 1080-5.

Yip, K.M. & Leng, G. 1998. Stability analysis for inverse simulation of aircraft. *Aeronautical Journal*, **102**(1016), 345-351.

Yue, A. & Postlethwaite, I. 1990. Improvement of helicopter handling qualities using H infinity optimisation. *IEE Proceedings: Control Theory and Applications*, **137**(3), 115-129.

Zames, G. 1981, Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control,* **26**(2), 301-320.

Zhao, Y. & Jayasuriya, S. 1994. Feedforward controllers and tracking accuracy in the presence of plant uncertainties. *Proceedings of the American Control Conference*; Jun 29th-Jul 1st, Baltimore, MD, US, 360-4.

Zhou, K., Doyle J.C., & Glover K. 1996. *Robust and optimal control*. Upper Saddle River, NJ: Prentice-Hall, In.

Zou, Q. & Devasia, S. 1999. Preview-based stable-inversion for output tracking. *Proceedings of the American Control Conference*; Jun 2-4, San Diego, CA, US, 3544-8.

Zou, Q. & Devasia, S. 2007. Preview-based inversion of nonlinear nonminimum-phase systems: VTOL example. *Automatica*, **43**(1), 117-127.

Appendix-A

Vector Relative Degree

This appendix contains an introduction of the definition and concepts associated with the vector relative degree for a MIMO square system (Isidori, 1989; Sastry, 1999). Here assume a nonlinear MIMO square system can be represented by the following form:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}) \cdot \boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{A.1}$$

where $\boldsymbol{u} \in \mathbb{R}^q$ is the input vector, $\boldsymbol{y} \in \mathbb{R}^q$ is the output vector, and $\boldsymbol{x} \in \mathbb{R}^m$ is the state variable vector. The variables $\boldsymbol{f}$, $\boldsymbol{g}$, and $\boldsymbol{h}$ are the function matrices with the corresponding orders.

Now differentiate the ith channel of the output vector $\boldsymbol{y}$ of Eq. (A.1) with respect to time to obtain the following equation:

$$\dot{\boldsymbol{y}}_i(t) = L_f \boldsymbol{h}_i(\boldsymbol{x}) + \sum_{j=1}^{q} [L_{g_j} \boldsymbol{h}_i(\boldsymbol{x})] \boldsymbol{u}_j \tag{A.2}$$

If the term $L_{g_j} \boldsymbol{h}_i(\boldsymbol{x})$ in Eq. (A.2) is equal to zero, the input $\boldsymbol{u}_j$ will not appear in this equation. Therefore, differentiate Eq. (A.2) further until one of the inputs appears in the final equations, as follows:

$$\dot{\boldsymbol{y}}_i^{(r_i)}(t) = L_f \boldsymbol{h}_i(\boldsymbol{x}) + \sum_{j=1}^{q} \{L_{g_j} [L_f^{r_i - 1} \boldsymbol{h}_i(\boldsymbol{x})]\} \boldsymbol{u}_j \tag{A.3}$$

where $\boldsymbol{r}_i$ is the smallest integer such that $L_{g_j} [L_f^{r_i - 1} \boldsymbol{h}_i(\boldsymbol{x})] \neq 0$, for some $\boldsymbol{x}$. If all the output channels are involved, the following compact equation can be obtained

$$\boldsymbol{y}^{(r)}(t) = \begin{bmatrix} \boldsymbol{y}_1^{r_1} \\ \vdots \\ \boldsymbol{y}_q^{r_q} \end{bmatrix} = \begin{bmatrix} L_f^{r_1}\boldsymbol{h}_1 \\ \vdots \\ L_f^{r_q}\boldsymbol{h}_q \end{bmatrix} + \boldsymbol{A}(\boldsymbol{x}) \begin{bmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_q \end{bmatrix} \tag{A.4}$$

where $\boldsymbol{A}(\boldsymbol{x})$ is

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} L_{\boldsymbol{g}_1}(L_f^{r_1-1}\boldsymbol{h}_1) & \cdots & L_{\boldsymbol{g}_q}(L_f^{r_q-1}\boldsymbol{h}_1) \\ \vdots & \ddots & \vdots \\ L_{\boldsymbol{g}_1}(L_f^{r_q-1}\boldsymbol{h}_q) & \cdots & L_{\boldsymbol{g}_q}(L_f^{r_q-1}\boldsymbol{h}_q) \end{bmatrix} \tag{A.4}$$

If the following equation

$$L_{\boldsymbol{g}_j}[L_f^{r_i-1}\boldsymbol{h}_i(\boldsymbol{x})] = 0 \qquad 0 \le k \le \boldsymbol{r}_i - 2 \text{ and } i = 1, \cdots, q \tag{A.5}$$

is satisfied in a neighbourhood of $\boldsymbol{x}_0$ and, in addition, the matrix $\boldsymbol{A}(\boldsymbol{x}_0)$ is nonsingular, then the system represented in Eq. (A.1) is said to have a vector relative degree $\boldsymbol{r} = [\boldsymbol{r}_1, \quad \boldsymbol{r}_2, \quad \cdots \quad \boldsymbol{r}_q]^T$.

It is thus apparent that the vector relative degree $\boldsymbol{r}$ is exactly equal to the number of times that the output $\boldsymbol{y}(t)$ has to be differentiated with respect to time to explicitly show the input $\boldsymbol{u}$ in the equation.

Appendix-B

HS125 (Hawker 800) Business Jet

This appendix provides a description of the nonlinear longitudinal mathematical model of a fixed-wing aircraft, the HS125 (Hawker 800) business jet (Thomson, 2004) which forms the basis of some of the work presented in Chapter 3 and Chapter 5. The vehicle equations of motion are:

$$\begin{aligned}
\dot{u} &= -qw + \frac{X}{m} - g\sin\theta \\
\dot{w} &= -qu + \frac{Z}{m} + g\cos\theta \\
\dot{q} &= \frac{M}{I_{yy}} \\
\dot{\theta} &= q \\
\dot{x}_e &= u\cos\theta - w\sin\theta \\
\dot{z}_e &= u\sin\theta + w\cos\theta
\end{aligned} \tag{B.1}$$

where u, w are velocity components in body axes, q is pitch angular velocity component in body axes, and θ is the fuselage-pitch attitude. X, Z, and M are the external forces and moment. The variables x_e and z_e are the longitudinal and vertical displacements, respectively.

Table B.1 provides details of the meaning of each of the parameters of this model and the values used.

Table B.1 Configuration data for the HS125 (Hawker 800) Business Jet

Parameter	Symbol	Value	Units
Aircraft Mass	M	7500	kg
Wing Area	S	32.8	m^2
Pitch Moment of Inertia	I_{yy}	84309	$Kg{\cdot}m^2$
Mean Chord	$\bar{c}$	2.29	m
Thrustline above x-axis	h_T	0.378	m
Lift Coefficients	C_{L0}, $C_{L\alpha}$, $C_{L\delta e}$	0.895,5.01,0.722	-
Drag Coefficients	C_{D0}, $C_{D\alpha}$, $C_{D\alpha\alpha}$	0.177,0.232,1.393	-
Pitching Moment Coefficients.	C_{M0}, $C_{M\alpha}$, $C_{M\delta e}$, C_{Mq}	-0.046,-1.087,-1.88,-7.055	-

In addition, this aircraft trim point at sea level with a velocity of 120 knots can be calculated as follows:

$$\begin{aligned} u_0 &= 61.8682 \text{ m/s}, \; w_0 = 0.8501 \text{ m/s}, \; q_0 = 0 \text{ rad/s} \\ \theta_0 &= 0.01374 \text{ rad/s}, x_{e_0} = 0.0 \text{ m}, z_{e_0} = 0.0 \text{ m} \end{aligned} \tag{B.2}$$

where the "0"subscript indicates that all these values are in a trimmed equilibrium flight state. The linearization of the dynamics around the equilibrium point are as follows:

$$\boldsymbol{A} = \begin{bmatrix} -0.0573 & 0.1243 & 0 & -9.8091 \\ -0.3051 & -0.8647 & 61.8682 & -0.1348 \\ -0.0015 & -0.0367 & -0.5455 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix} \tag{B.3}$$

$$\boldsymbol{B} = \begin{bmatrix} 0.1019 & 0.0001 \\ -7.4188 & 0 \\ -3.9275 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.4}$$

Appendix-C

Inverse Identification

In this appendix the sensitivity-analysis technique is used to determine the key parameters in the obtained model, and then the traditional identification method is implemented to identify these key parameters in the inverse structure.

All the introduced methods in the main context of the thesis, including model inversion and inverse simulation, are based on the pure mathematical method to find the desired inputs. The inverse identification presented in this appendix is concerned with identifying an inverse model from the input and output data sets. This technique is combined with the approaches for model inversion in that it utilises the structure of the inverse model. Besides, the input vector is constructed according to the relative order $\boldsymbol{r}$. However, if the component value of $\boldsymbol{r}$ is larger than two, in reality, it will lead to the difficulty in constructing the input vector. This is due to the fact that the real data is often polluted by noise and its signal-to-noise ratio (SNR) is not high. Therefore, before applying the approach to real data, it is necessary to reprocess the data to select the frequency range of interest etc. In addition, the sensitivity analysis method shown in the paper (Sato *et al*., 2004) is used to determine the key parameters to improve the identification process.

Sensitivity analysis is relatively fast and computationally inexpensive but it only describes the local behaviour of the model outcome uncertainty as results of the model parameter uncertainties (Haverkort & Meeuwissen, 1995). Moreover, the inverse model is derived from the forward model which is valid in the local area of the equilibrium point. So these

parameters will be only valid in this local area. This coincidently is consisted with the necessary condition for the application of the sensitivity analysis method. Furthermore, sensitivity analysis will face difficulty if the parameters are dependent or coupled with each other. This is a common situation in the aircraft model especially for the helicopter model. Fortunately, here the considered model is the noncausal inverse model of the original model. These parameters lack physical meaning and this consequently leads to a weak dependent relationship among them.

A series of tests with simulated data have been run on the Autogyro longitudinal model (Houston, 1998). The vector relative degree $\boldsymbol{r}$ of this system is [1, 1, 1, 2, 1]. Based on this value, it is possible to construct the input vector for the inverse model. The ideal system matrices of the inverse model are shown as following.

$$\boldsymbol{A} = \begin{bmatrix} 0.0403 & -0.2880 & -4.1061 & -9.0035 & -0.0407 \\ 0.3473 & 0.8317 & 24.9658 & 1.9215 & -0.0869 \\ -0.0951 & -0.5347 & -1.5902 & -0.3546 & 0.0085 \\ 0 & 0 & 1.0000 & 0 & 0 \\ 0.1306 & 0.3838 & 8.6225 & 0.9926 & -0.0288 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 0.0010 & -0.0095 & 0.0028 & 0.0028 & 0.0291 \\ -0.0095 & 0.0954 & -0.0277 & -0.0277 & -0.2910 \\ 0.0028 & -0.0277 & 0.0080 & 0.0080 & 0.0844 \\ 0 & 0 & 0 & 0 & 0 \\ 0.0291 & -0.2910 & 0.0844 & 0.0844 & 0.8876 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} -4.0733 & -16.1973 & -13.0081 & 3.2546 & 0.328 \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} 0.0954 & -0.9541 & 0.2767 & 0.2767 & 2.9101 \end{bmatrix}$$

Now sensitivity analysis is used to determine the key parameters. The results show that the key parameters for $\boldsymbol{A}$ are: $\boldsymbol{A}(1,1)$, $\boldsymbol{A}(1,5)$, $\boldsymbol{A}(2,1)$, $\boldsymbol{A}(2,5)$, $\boldsymbol{A}(3,1)$, $\boldsymbol{A}(3,2)$, $\boldsymbol{A}(3,5)$, $\boldsymbol{A}(4,1)$, $\boldsymbol{A}(4,2)$, $\boldsymbol{A}(4,3)$, $\boldsymbol{A}(4,4)$, $\boldsymbol{A}(4,5)$, $\boldsymbol{A}(5,1)$; The same approach is used for the $\boldsymbol{B}$ matrix giving: $\boldsymbol{B}(1,1)$, $\boldsymbol{B}(1,5)$, $\boldsymbol{B}(2,1)$, $\boldsymbol{B}(2,5)$, $\boldsymbol{B}(3,1)$, $\boldsymbol{B}(3,2)$, $\boldsymbol{B}(3,5)$, $\boldsymbol{B}(4,1)$, $\boldsymbol{B}(4,2)$, $\boldsymbol{B}(4,5)$ as key parameters.

The final results have shown the validity of the method. Two sets of parameters have been identified by keeping the other parameters at ideal values. Since lacking of the physical relationship among the parameters, the components of these two sets are selected randomly. The results are presented in Table C.1 to Table C.6 and also show the effect of the increasing noise level. Compared with the ideal values, the identified results show good results. The first set selects four components of the ***A*** matrix and two components of the ***B*** matrix. In Table C.1, all the parameters obtain well-identified values except the component ***B***(2,1) without noise. However it retains the same negative sign as the ideal value. When the SNR is reduced, the ***B***(2,1) values quickly turns its negative sign to positive. Other parameters show greater robustness to noise. The second set selects three components each from the ***A*** and ***B*** matrices. For this case nearly all the parameters are estimated well shown in the Table C.4 to Table C.6. Although the ***B***(1,1) value changes fast with the reduction of the SNR, it still keeps the same positive sign.

In addition, the method has also been run on twelve parameters, the combination of the above two sets. The results have shown that without noise, all the parameters obtain relatively good results except the parameters ***B***(1,1) and ***B***(2,1). They also have not been identified well in the previous investigations. The difference from the above results is that here the identified results quickly deteriorate when the noise level is increased. This is probably due to too many parameters are to be identified at one time.

Table C.1 Parameter identification of the inverse Autogyro longitudinal model (Noise level=0 for each channel)

Components	Ideal value	Guess value	Estimated
A(1,1)=	0.0403	0.03	0.0364
A(1,5)=	-0.0407	-0.06	-0.0315
A(2,1)=	0.3473	0.5	0.3583
A(2,5)=	-0.0869	-0.5	-0.1008
B(2,1)=	-0.0095	-0.007	-0.0374
B(3,5)=	0.0844	0.1	0.0849

Table C.2 Parameter identification of the inverse Autogyro longitudinal model (Noise level=0.00001 for each channel)

Components	Ideal value	Guess value	Estimated
A(1,1)=	0.0403	0.03	0.0316
A(1,5)=	-0.0407	-0.06	-0.025
A(2,1)=	0.3473	0.5	0.3404
A(2,5)=	-0.0869	-0.5	-0.1026
B(2,1)=	-0.0095	-0.007	0.0105
B(3,5)=	0.0844	0.1	0.0855

Table C.3 Parameter identification of the inverse Autogyro longitudinal model (Noise level=0.0001 for each channel)

Components	Ideal value	Guess value	Estimated
A(1,1)=	0.0403	0.03	0.0369
A(1,5)=	-0.0407	-0.06	-0.0096
A(2,1)=	0.3473	0.5	0.2969
A(2,5)=	-0.0869	-0.5	-0.1068
B(2,1)=	-0.0095	-0.007	0.0894
B(3,5)=	0.0844	0.1	0.0865

Table C.4 Parameter identification of the inverse Autogyro longitudinal (Noise level=0 for each channel)

Components	Ideal value	Guess value	Estimated
A(3,1)=	-0.0951	-0.1	-0.0959
A(3,5)=	0.0085	0.005	0.0085
A(5,1)=	0.1306	0.1	0.1408
B(1,1)=	0.0010	0.006	0.0081
B(1,5)=	0.0291	0.01	0.0265
B(3,2)=	-0.0277	-0.04	-0.0257

Table C.5 Parameter identification of the inverse Autogyro longitudinal model (Noise level=0.00001 for each channel)

Components	Ideal Value	Guess Value	Estimated
A(3,1)=	-0.0951	-0.1	-0.0973
A(3,5)=	0.0085	0.005	0.0074
A(5,1)=	0.1306	0.1	0.2003
B(1,1)=	0.0010	0.006	0.053
B(1,5)=	0.0291	0.01	0.0494
B(3,2)=	-0.0277	-0.04	-0.0273

Table C.6 Parameter identification of the inverse Autogyro longitudinal model (Noise level=0.001 for each channel)

Components	Ideal value	Guess value	Estimated
A(3,1)=	-0.0951	-0.1	-0.1021
A(3,5)=	0.0085	0.005	0.0057
A(5,1)=	0.1306	0.1	0.3226
B(1,1)=	0.0010	0.006	0.173
B(1,5)=	0.0291	0.01	0.093
B(3,2)=	-0.0277	-0.04	-0.0298

Appendix-D

The Nonlinear Norrbin Ship Model

This appendix provides information concerning the nonlinear Norrbin ship model. In the autopilot design field, the Norrbin model has been used extensively for ship manoeuvring studies involving both deep and confined waters (Fossen, 1994; Unar, 1999). In this thesis, this model is used to describe the motion of the RZ ship, which will be considered as a design benchmark to investigate the inverse simulation technique and the effectiveness of the feedback controller designed by the mixed-sensitivity (*K*/*KS*) H_∞ algorithm. The structure of the Norrbin ship model can be represented by the following equation:

$$T\ddot{\Psi} + H_N(\dot{\Psi}) = K\delta \tag{D.1}$$

where δ and Ψ represents the rudder and heading angles, respectively. *K* and *T* are constants and the nonlinear term $H_N(\dot{\Psi})$ is defined as:

$$H_N(\dot{\Psi}) = \alpha_3\dot{\Psi}^3 + \alpha_2\dot{\Psi}^2 + \alpha_1\dot{\Psi} + \alpha_0 \tag{D.2}$$

where α_i ($i = 0,1,2,3$) are called Norrbin's coefficients. For most ships, α_3 = α_2 = 0. Therefore, Eqs. (D.1) and (D.2) can be simplified to form Eq. (D.3):

$$\delta = m\ddot{\Psi} + d_1\dot{\Psi} + d_2\dot{\Psi}^3 \tag{D.3}$$

where $m = \frac{T}{K}$, $d_1 = \frac{\alpha_1}{K}$, and $d_3 = \frac{\alpha_3}{K}$. These coefficients will vary according to chosen operating point – in this case involving the changing forward speed *U*. The coefficient values correspond to the forward speeds from 1 m/s to 20 m/s can be found in Table D.1. Now Eq. (D.3) can be transformed into a state-space form, as shown in Eq. (D.4), for facilitating the investigation of inverse simulation.

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{d_1}{m}x_2 - \frac{d_3}{m}x_2^3 + \frac{1}{m}\delta \\ \dot{\delta} &= \frac{1}{\tau}(\delta_c - \delta)\end{aligned} \tag{D.4}$$

where $x_1 = \Psi$ and τ is the time constant. The third equation in Eq. (D.4) is related to the steering machine structure, as described in Fig. 1, where the rudder and rudder rate limiters are involved in the model.

Table D.1 Parameter variations with respect to *U* (Unar, 1999)

***U*(m/s)**	***T***	***K***	***m* = *T*/*K***	$d_1 = \alpha_1/K$	$d_3 = \alpha_3/K$
1	155.0000	0.1	1550.0000	10.0000	100.0000
2	77.5000	0.2	387.5000	5.0000	12.5000
3	51.6667	0.3	172.2222	3.3330	3.7037
4	38.7500	0.4	96.8750	2.5000	1.5625
5	31.0000	0.5	62.0000	2.0000	0.8000
6	25.8333	0.6	43.0556	1.6667	0.4630
7	22.1429	0.7	31.6327	1.4286	0.2915
8	19.3750	0.8	24.2188	1.2500	0.1953
9	17.2222	0.9	19.1358	1.1111	0.1372
10	15.5000	1.0	15.5000	1.0000	0.1000
11	14.0909	1.1	12.8099	0.9091	0.0751
12	12.9167	1.2	10.7639	0.8333	0.0579
13	11.9231	1.3	9.1716	0.7692	0.0455
14	11.0714	1.4	7.9082	0.7143	0.0364
15	10.3333	1.5	6.8889	0.6667	0.0296
16	9.6875	1.6	6.0547	0.6250	0.0244
17	9.1176	1.7	5.3633	0.5882	0.0204
18	8.6111	1.8	4.7840	0.5556	0.0171
19	8.1579	1.9	4.2936	0.5263	0.0146
20	7.7500	2.0	3.8750	0.5000	0.0125